VEB RECHENELEKTRONIK
MEININGEN/ZELLA-MEHLIS
IM VEB KOMBINAT
ZENTRONIK

PROGRAMMIERUNG UND BEDIENUNG DES DIGITALEN KLEINRECHNERS C 8205

HEFT 5
FESTKOMMA-INTERPRETIERSYSTEM FIPS I

Nachdruck, Reproduktionen und Nachbildung – auch auszugsweise – nur mit Genehmigung des Herstellerbetriebes gestattet.

Programmierung und Bedienung

des digitalen Kleinrechenautomaten

C 8205

Heft 5

Festkommainterpretiersystem FIPS I

Inhaltsverzeichnis

		Seite
0.	**Einleitung**	**5**
1.	**Informationsdarstellungen**	**6**
1.1.	Triadendarstellung	6
1.2.	Zahlendarstellung	6
1.3.	Befehlsdarstellung	7
1.3.1.	Adressierung	7
1.3.2.	Externbefehle	8
1.4.	Textdarstellung	8
2.	**Eingabeorganisation**	**9**
2.1.	Relative Programmierung	9
2.2.	Eingabeprogramm	9
2.2.1.	Hilfsadressen	10
2.2.2.	Abzuspeicherndes Maschinenwort	11
2.2.3.	Sofort-Maschinenbefehl	12
2.2.4.	Abzuspeichernder Externbefehl	13
2.2.5.	Sprungbefehl extern	13
2.2.6.	Dezimalzahl	14
2.2.7.	Dezimalzahlen im Satz	15
2.2.8.	Wortmarke	15
2.2.9.	Lochbandschlußzeichen	15
2.2.10.	Zwischenraum	15
2.2.11.	Programme mit B- und D-Taste	15
3.	**Externbefehle**	**16**
3.1.	Indexregister und Indexmodifikation	16
3.2.	Erläuterung der Externbefehle	16
3.2.1.	Lesebefehl	16
3.2.2.	Transportbefehl	16
3.2.3.	Lesebefehl, negativ	17
3.2.4.	Löschbefehl	17
3.2.5.	Indexregister-Lesebefehl	17
3.2.6.	Indexregister-Transportbefehl	18
3.2.7.	Indexregister-Lesebefehl, negativ	18
3.2.8.	Indexregister-Löschbefehl	18
3.2.9.	Addition	18
3.2.10.	Subtraktion	19
3.2.11.	Konjunktion, binär	19
3.2.12.	Vergleichen	20
3.2.13.	Indexregister-Addition	20
3.2.14.	Indexregister-Subtraktion	20
3.2.15.	Indexregister-Konjunktion, binär	20
3.2.16.	Indexregister-Vergleichen	21
3.2.17.	Betragsbildung	21
3.2.18.	Verschiebung links	21
3.2.19.	Verschiebung rechts	22
3.2.20.	Multiplikation	23
3.2.21.	Division	23
3.2.22.	Konjunktion dezimal	24
3.2.23.	Unbedingter Sprung	25
3.2.24.	Vergleichersprung 1	25
3.2.25.	Vergleichersprung 2	26
3.2.26.	Bedingter Sprung – Registerinhalt negativ	26
3.2.27.	Bedingter Sprung – Registerinhalt gleich Null	26
3.2.28.	Bedingter Sprung – Registerinhalt nicht negativ	26
3.2.29.	Bedingter Sprung – Indexregisterinhalt negativ	27
3.2.30.	Bedingter Sprung – Indexregisterinhalt gleich Null	27
3.2.31.	Bedingter Sprung – Indexregisterinhalt nicht negativ	27
3.2.32.	Überlaufsprung	28
3.2.33.	Selektorsprung	28

Seite
3.2.34. Bedingter Stop 29
3.2.35. Unbedingter Stop 29
3.2.36. Anwahl Kanal Null 29
3.2.37. Anwahl Kanal Eins 29
3.2.38. Anwahl Kanal Zwei 29
3.2.39. Eingabe einer Dezimalzahl 30
3.2.40. Eingabe von Dezimalzahlen im Satz 30
3.2.41. Eingabe eines Wortes mit Kennzeichen 31
3.2.42. Eingabe eines Programmteiles 32
3.2.43. Wagenrücklauf 33
3.2.44. Wagenrücklauf mit Zeilenvorschub 33
3.2.45. Zeilenvorschub 33
3.2.46. Zwischenraum 33
3.2.47. Tabulator 34
3.2.48. Ausgabe einer Dezimalzahl/Variante 1 34
3.2.49. Ausgabe einer Dezimalzahl/Variante 2 35
3.2.50. Ausgabe einer Dezimalzahl/Variante 3 35
3.2.51. Ausgabe von Triaden mit Eingabekennzeichen 36
3.2.52. Textausgabe 37
3.2.53. Unterprogrammsprung 38

4. Programmierung in der Maschinensprache 40

4.1. Eingabeprogramme 40
4.1.1. Eingabe einer Dezimalzahl 40
4.1.2. Eingabe von Dezimalzahlen im Satz 40
4.1.3. Eingabe eines Wortes mit Kennzeichen 41
4.1.4. Eingabe eines Programmteiles 41
4.2. Rechenprogramme 41
4.2.1. Verschiebung links 42
4.2.2. Verschiebung rechts 42
4.2.3. Multiplikation 42
4.2.4. Division 43
4.2.5. Konjunktion dezimal 43
4.3. Ausgabeprogramme 43
4.3.1. Ausgabe einer Dezimalzahl/Variante 1 44
4.3.2. Ausgabe einer Dezimalzahl/Variante 2 44
4.3.3. Ausgabe einer Dezimalzahl/Variante 3 44
4.3.4. Ausgabe von Triaden mit Eingabekennzeichen 44
4.3.5. Textausgabe 45

5. Zusammenhang zwischen Extern- und Maschinenbefehlen 46

6. Hilfsprogramme 48

6.1. Druck von Pseudozähler und Inhalt des Registers R 48
6.2. Fortsetzung eines unterbrochenen, interpretierend arbeitenden Programmteiles 48
6.3. Druck des Inhaltes der Zelle 7777 49
6.4. Bereichsausschreiben 49
6.5. Bereichsausstanzen 49
6.6. Vergleichsprogramm 50

7. Anfertigen eines Lochbandes und Korrektur von Eingabefehlern 51

7.1. Anfertigen eines Lochbandes und Korrektur von Lochfehlern 51
7.2. Eingabe über Tastatur 52

8. Bedienungshinweise 53

8.1. Ansprung des „Eingabeprogramms" 53
8.2. Programmeingabe 53
8.3. Korrektur von fehlerhaften Plätzen 53
8.4. Druck des Maschinenbefehlszählers und des Maschinenakkumulators 54
8.5. Druck des Inhaltes einer Zelle 54

0. Einleitung

Das Rechenwerk des C 8205 ermöglicht nur die Ausführung bestimmter Elementaroperationen, wie z. B. die Festkommaaddition und -subtraktion, die binäre Verschiebung, die bitweise Konjunktion oder die Ein- und Ausgabe eines Zeichens.

Es ist daher notwendig, Operationen wie z. B. die Ein- und Ausgabe von Befehlen und Zahlen oder die Festkommamultiplikation bzw. -division durch spezielle Programme zu realisieren. Diese können als sogenannte „offene Unterprogramme" bereitgestellt und durch einen entsprechenden Maschinenbefehl aufgerufen werden.

Da die Versorgung der offenen Unterprogramme vom Objektprogramm durchzuführen ist, kann sich für dieses ein relativ hoher Speicherplatzbedarf ergeben.

Das vorliegende Interpretiersystem ermöglicht die Aufstellung speichergünstiger Objektprogramme. Durch die Realisierung von sogenannten „Externbefehlen" soll dem Programmierer die Möglichkeit gegeben werden, ohne umfassende Kenntnis der inneren Struktur des Automaten Aufgaben auf der Grundlage der Festkommarechnung bei einfacher Genauigkeit zu lösen.

Mit Hilfe der Externbefehle wird im wesentlichen ein Einadreßautomat mit einem Rechenregister R und sieben Indexregistern I_1 bis I_7 simuliert. Der Befehlsaufruf erfolgt durch einen Pseudozähler PZ. Die Externbefehle werden interpretierend abgearbeitet; sie werden erst nach ihrem Aufruf durch ein spezielles Programm entschlüsselt.

Die im vorliegenden System enthaltenen Externbefehle werden für die Lösung vieler Probleme ausreichen. Darüber hinaus hat der Programmierer die Möglichkeit, selbst Befehle zu realisieren, die für seine spezielle Aufgabe wirkungsvoller sind.

Ist die Aufstellung zeitgünstigerer Programmteile oder Programme notwendig, muß auf die Maschinenbefehle bzw. auf entsprechende Interpretiersysteme zurückgegriffen werden.

Das vorliegende System ermöglicht einen wahlweisen Übergang zwischen der interpretierenden Abarbeitung von Extern- und Maschinenbefehlen und der direkt verdrahteten Verarbeitung von Maschinenbefehlen. Die Externbefehle können somit auch als Ergänzung der verdrahteten Befehle angesehen werden.

Das Interpretiersystem „FIPS I" belegt die ersten 32 Spuren des Speichers. Als Arbeitszellen des Systems werden die letzten 161 Zellen verwendet. Damit stehen diese nicht frei zur Verfügung.

Die Adressierung des wortorientierten Speichers erfolgt im vorliegenden Standardsystem im Oktalsystem. Durch einen Programmzusatz ist der Übergang zum Dezimalsystem möglich.

Das Rechenregister R, die Indexregister I_1 bis I_7 und der Pseudozähler PZ sind adressierbar.

Die Externbefehle werden ausführlich beschrieben. Bei den Kapiteln, die sich auf die Programmierung in der Maschinensprache beziehen, wird die Kenntnis des Heftes 2 „Anleitung zur Programmierung in der Maschinensprache" vorausgesetzt.

Für die Testung von Programmen werden Hilfsprogramme zur Verfügung gestellt, die an beliebig freie Speicherplätze gebracht werden können.

Bei der Abarbeitung erprobter Programme werden sie nicht benötigt.

Die Ausführungen in diesem Heft beziehen sich auf die 8- und 5-Kanal-Variante (Externer R 300-Code bzw. Fernschreibcode Nr. 2). Auf Unterschiede in den beiden Versionen wird an den entsprechenden Stellen hingewiesen.

1. Informationsdarstellungen

Der Speicher des C 8205 ist in 128 Spuren zu je 32 Sektoren aufgeteilt.

Die kleinste adressierbare Informationseinheit ist das Wort. Es stellt den Inhalt einer Zelle dar (letztere wird im weiteren Verlauf auch als Speicherplatz bezeichnet). Eine Information wird intern rein binär dargestellt. Ob eine solche Bit-Information als Zahl, alphanumerisches Zeichen oder Befehl interpretiert wird, hängt von der Art der Operation ab, der ein Speicherplatz unterzogen wird.

1.1. Triadendarstellung

Jedes Wort besteht, unabhängig von seiner speziellen Bedeutung, aus 33 Bits, die dezimal mit 0 bis 32 von links nach rechts durchnumeriert sind (Abb. 1). Jede dieser 33 Dualstellen kann mit den Ziffern „0" oder „1" besetzt sein, so daß es insgesamt 2^{33} verschiedene Maschinenworte gibt. Bei jedem Maschinenwort kann man, um zu einer kürzeren Schreibweise zu gelangen, je drei Dualziffern zu einer Oktalziffer zusammenfassen. Die Zuordnung und Bezeichnung der Wortstellen ist in Abb. 1 angegeben. Außerdem ist in dieser Abbildung ein Maschinenwort in dualer und oktaler Schreibweise (Triadendarstellung) gegenübergestellt.

Bezeichnung oktal	t_{10}			t_9			t_8			t_7			t_6			t_5			t_4			t_3			t_2			t_1			t_0		
Bezeichnung dual	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}	z_{11}	z_{12}	z_{13}	z_{14}	z_{15}	z_{16}	z_{17}	z_{18}	z_{19}	z_{20}	z_{21}	z_{22}	z_{23}	z_{24}	z_{25}	z_{26}	z_{27}	z_{28}	z_{29}	z_{30}	z_{31}	z_{32}
Maschinenwort dual	0	1	1	1	0	1	0	0	0	1	1	1	0	1	0	1	1	0	0	1	1	1	0	1	0	0	1	1	0	0	1	1	1
Maschinenwort oktal	3			5			0			7			2			6			3			5			1			4			7		

Abb. 1

1.2. Zahlendarstellung

Die Befehle des Systems „FIPS I" ermöglichen eine Verarbeitung von Festkommazahlen. Die „I" weist darauf hin, daß zur Darstellung einer Festkommazahl ein Maschinenwort verwendet wird (einfache Genauigkeit).

Ein Komma wird nicht wirklich gesetzt, sondern der Programmierer muß eine gedachte Kommastellung bei der Aufstellung seines Programmes verfolgen.

Die Festkommazahlen werden intern rein binär verschlüsselt. Für die Bezeichnung der Dualstellen gilt die in Abb. 1 angegebene Anordnung. Die Dualstelle z_{32} wird zur Darstellung einer Festkommazahl nicht benutzt. Sie bleibt frei.

Positive ganze Zahlen (Komma zwischen z_{31} und z_{32}) werden durch

$$Z = \sum_{i=1}^{31} z_i \, 2^{31-i}$$

beschrieben.

Die der Dualstelle z_i zugeordnete Zweierpotenz beträgt 2^{31-i} ($i = 1, 2, \ldots, 31$). Die Einheit der größten Stelle ist 2^{30}, die der kleinsten 2^0.

Negative Zahlen werden unter Berücksichtigung der Wortlänge von 32 Bits im Zweierkomplement dargestellt (Beispiel 1). Damit sind negative Zahlen intern durch $z_0 = 1$, nicht negative durch $z_0 = 0$ charakterisiert. Für ganze Zahlen Z ergibt sich intern ein Zahlenbereich von

$$|Z| \leq 2^{31} - 1 = 2\ 147\ 483\ 647\ ^{1)}$$

[1]) Die intern noch darstellbare Zahl $Z = -2^{31}$ (nur z_0 besetzt) führt bei den Operationen Dezimalverschiebung, dezimale Konjunktion, Dezimalausgaben, Multiplikation und Division von Dezimalzahlen zum Überlauf.

Dezimalzahlen werden bei ihrer Eingabe in den Automaten durch ein entsprechendes Leseprogramm in das Dualsystem überführt. Der Eingabe-Zahlenbereich stimmt mit dem internen überein. Die möglichen 10 Dezimalstellen werden mit $d_0, d_1, \ldots, d_9$ bezeichnet, wobei d_0 die niedrigste Dezimalstelle ist.

$$|D| = |d_9\, d_8 \ldots d_0| \leqq 2\ 147\ 483\ 647$$

Für d_9 ergibt sich entsprechend dem Zahlenbereich die Beschränkung $d_9 \leqq 2$. Die Ein- und Ausgabedarstellungen von Dezimalzahlen sind in den Abschnitten 2.2.6. und 2.2.7. bzw. bei der Erläuterung der entsprechenden Befehle angegeben (siehe Abschnitt 3.2.).

Wenn man die interne Darstellung einer ganzen positiven Dezimalzahl in der Triadenschreibweise zusammenfaßt, erhält man die zugeordnete Oktalzahl, oktal multipliziert mit 2.

Beispiel 1: Der internen Darstellung „00 ... 0110010" entspricht die Dezimalzahl 25.

Der Dezimalzahl –25 entspricht die interne Darstellung „11 ... 11001110". Man erhält letztere durch Umrechnen der Dezimalzahl 25 ins Dualsystem, Verschiebung um eine Dualstelle nach links, stellenweiser Negation und Addition einer 1 in der Dualstelle z_{32}.

1.3. Befehlsdarstellung[2])

1.3.1. Adressierung

Zur Adressierung der dezimal 4096 Speicherplätze werden intern 12 Bits benötigt ($2^{12} = 4096$). Diese werden für eine kürzere äußere Darstellung analog der Triadenschreibweise zu 4 Oktalziffern $a = a_1a_2a_3a_4$ zusammengefaßt. Damit ergibt sich für die dezimal 4096 Speicherplätze im Oktalsystem eine durchgehende Numerierung von 0 bis 7777 (dezimal 0 bis 4095).

Für die oktale Adressierung sind keine weiteren Kenntnisse über das Oktalsystem erforderlich, wenn man nur beachtet, daß im Oktalsystem die Dezimalziffern 8 und 9 nicht auftreten.

Beispiel 2: Durchgehende Numerierung im Oktalsystem:

... 5, 6, 7, 10, 11, ...
... 16, 17, 20, 21, 22, ...
... 76, 77, 100, 101, 102, ...
... 1776, 1777, 2000, 2001, ...

Durch die Angabe ihrer Adresse a ist eine Zelle eindeutig bestimmt, so daß im weiteren Verlauf auch einfach von der Zelle a bzw. dem Speicherplatz a oder auch einfach von „a" gesprochen wird.

Das System „FIPS I" belegt oktal die Speicherplätze 0 bis 1777. Die Zellen 7537 bis 7777 stehen für den Programmierer nicht frei zur Verfügung, da sie vom System als Arbeitszellen verwendet werden.

Das Rechenregister R, der Pseudozähler PZ und die Indexregister I_1 bis I_7 sind bestimmte, fest definierte Zellen des Hauptspeichers und daher adressierbar:

R ist oktal die Adresse 7773 zugeordnet
PZ ist oktal die Adresse 7751 zugeordnet
I_1 ist oktal die Adresse 7741 zugeordnet
I_2 ist oktal die Adresse 7742 zugeordnet
.
.
I_7 ist oktal die Adresse 7747 zugeordnet

[2]) Die Darstellung eines Maschinenbefehls wird nicht erläutert. Sie ist in der „Anleitung zur Programmierung in der Maschinensprache" angegeben.

1.3.2. Externbefehle

Durch die Externbefehle wird im wesentlichen ein Einadreßautomat mit 7 Indexregistern simuliert, obwohl eine bestimmte Gruppe von Externbefehlen praktisch die Wirkung von Zweiadreßbefehlen hat (Verknüpfung zwischen Indexregister und einer Zelle des Hauptspeichers). Darüber hinaus gibt es Externbefehle, die wie (1 + 1) – Adreßbefehle wirken (z. B. Schreibwerkfunktionen mit anschließendem unbedingten Sprung).

Zur Darstellung eines Externbefehles werden intern die Bits z_0 bis z_{31} benutzt. Die Dualstelle z_{32} ist immer mit 1 besetzt. Die Bits z_0 bis z_5 enthalten den Operationscode, die Bits z_6 bis z_8 bezeichnen die Nummer des Indexregisters I_n ($n = 1, \ldots, 7$). Die 12 Bits z_9 bis z_{20} enthalten die Grundadresse des Befehls. Die Bits z_{21} bis z_{31} nehmen bei einer Gruppe von Befehlen einen Operationszusatz (z. B. Verschiebeanweisung) auf. Dieser kann max. 3 Parameter enthalten (Abb. 2).

Bemerkung: Auch bei Befehlen ohne Operationszusatz sind die Bits z_{21} bis z_{31} in der internen Darstellung besetzt.

Operationscode	Indexregister	Grundadr.	Operationszusatz			1
			3. Param.	2. Param.	1. Param.	
$z_0 \ldots z_5$	$z_6 \ldots z_8$	$z_9 \ldots z_{20}$	$z_{21} \ldots z_{23}$	$z_{24} \ldots z_{27}$	$z_{28} \ldots z_{31}$	z_{32}

Abb. 2

In der Eingabedarstellung eines Externbefehles wird der Operationscode durch zwei Oktalziffern S_1 und S_2, das Indexregister durch eine Oktalziffer I und die Grundadresse entsprechend der oktalen Adressierung durch 4 Oktalziffern $a = a_1a_2a_3a_4$ angegeben. Die Parameterangabe im Operationszusatz erfolgt dezimal.

Die Externbefehle werden bei ihrer Eingabe in den Automaten durch ein entsprechendes Leseprogramm in die interne Darstellung überführt. Dabei erfolgt auch eine Umschlüsselung des Operationscodes. Die Eingabedarstellung der Externbefehle ist im Abschnitt 2.2.4. bzw. 3.2. angegeben.

1.4. Textdarstellung

Für die Darstellung eines alphanumerischen Zeichens werden intern 6 Bits benötigt. Jedem Zeichen ist eine ganz bestimmte Bitkombination zugeordnet. In einem Wort können 5 alphanumerische Zeichen entsprechend der Abb. 3 gespeichert werden.

Eine numerische Eingabedarstellung der alphanumerischen Zeichen ergibt sich, wenn man die 6 Bits eines Zeichens zu zwei Oktalziffern (Triaden) zusammenfaßt. Diese Zuordnung ist in der Tabelle 1 angegeben.

Die 5 in einem Wort speicherbaren alphanumerischen Zeichen können in der Triadenschreibweise zusammengefaßt werden (Abb. 3).
Die Entschlüsselung geschieht durch ein spezielles Ausgabeprogramm. Der Aufruf dieses Programmes kann durch einen entsprechenden Externbefehl erfolgen (Abschnitt 3.2.52.).

Eine alphanumerische Eingabe ist im vorliegenden Standardsystem nicht enthalten. Sie kann über ein Zusatzprogramm realisiert werden.

	1. Z.		2. Z.		3. Z.		4. Z.		5. Z.	
t_{10}	t_9	t_8	t_7	t_6	t_5	t_4	t_3	t_2	t_1	t_0
bel.	011	111	010	100	101	111	000	100	010	001
bel.	3	7	2	4	5	7	0	4	2	1
	GB		D		KB		4		a	

Abb. 3

2. Eingabeorganisation

Die Eingabe von Informationen wird beim C 8205 durch spezielle Leseprogramme realisiert. Es muß dazu eine Möglichkeit geben, die es dem Automaten gestattet, die verschiedenen Informationen, die ihm mitgeteilt werden sollen, zu unterscheiden. Außerdem muß der Speicherbereich für die einzugebenden Informationen festgelegt werden. Dabei soll der Benutzer nicht bereits bei der Programmierung an einen bestimmten Speicherbereich gebunden sein (relative Programmierung).

Wenn die Auswahl eines speziellen Leseprogramms nicht durch einen entsprechenden Befehl, z. B. durch einen bestimmten Externbefehl, durchgeführt wird, erfolgt die Steuerung bei der Eingabe durch ein besonderes Eingabekennzeichen, dem sogenannten „1. Kennzeichen K_1".

Die Festlegung eines bestimmten Speicherbereiches bei der Eingabe erfolgt durch die Angabe von „Hilfsadressen" (Leitadressen und Relativadresse) und einem „2. Kennzeichen K_2".

2.1. Relative Programmierung

Die relative Programmierung ermöglicht es, ein aufgestelltes Programm auf einem beliebig freien Bereich des Speichers arbeiten zu lassen. Bei einem Befehl, der von diesem Speicherbereich abhängt, wird nicht die endgültige absolute Adresse eingetragen, sondern eine Grundadresse, die sich aus der relativen Stellung des einzugebenden Befehls ergibt.

Man bezieht sich bei der Adressierung eines Programms auf die Adresse Null. Befehle, deren Grundadresse von diesem Anfang abhängen, werden markiert. Bei der Eingabe eines Programms wird der Adreßteil der so markierten Befehle um die Anfangsadresse des gewünschten Speicherbereichs erhöht (Adressenänderung 1. Art). Diese Adressenänderung wird beim C 8205 mit Hilfe der sogenannten 1. Leitadresse (1. LA) durchgeführt.

Darüber hinaus gibt es beim „FIPS I" zur Adressenänderung 1. Art noch weitere 8 Hilfsadressen, die als 2., 3., . . ., 9. Leitadresse bezeichnet werden. Man benutzt sie zweckmäßigerweise als Anfangsadresse für Felder, die vom Programm benötigt werden, zur Adressenänderung der Sprungbefehle zu den einzelnen Unterprogrammen oder zur Parameterversorgung der Unterprogramme. Die Adressenänderung 1. Art wird mit Hilfe des 2. Kennzeichens K_2 durchgeführt.

2.2. Eingabeprogramm

Die Ausführungen in diesem Abschnitt beziehen sich auf diejenige Eingabeorganisation, die mit Hilfe des 1. Kennzeichens K_1 durchgeführt wird.

K_1 bestimmt die Interpretation bzw. Verarbeitung des einzulesenden Wortes. Einige Kennzeichen K_1 verlangen noch die Angabe des 2. Kennzeichens K_2. Die Eingabekennzeichen werden nicht gespeichert. Sie dienen nur zur Steuerung. Sie werden jeweils vor der eigentlichen Information gegeben.

Von den im Code – externer R 300-Code bzw. Fernschreibcode Nr. 2 – enthaltenen Zeichen werden als K_1 die

- Ziffern 0 | 1 | 2 | 3 | 4 | 5 | 8
- Zeichen ZR | . | : | / | Tab (8-Kanal) | BM (8-Kanal)
- Buchstaben B | D

verwendet.

Alle anderen Zeichen führen, wenn sie als 1. Kennzeichen vorkommen, zur Eingabe eines 1. Kennzeichens zurück.

Die einzugebenden Informationen werden im allgemeinen durch eine Wortmarke „WM" abgeschlossen. In der 8-Kanal-Version können als WM die beiden Zeichen Tabulator „Tab" oder Punkt „." verwendet werden, in der 5-Kanal-Version nur das Zeichen Punkt „.".

Bemerkung: Bei Beispielen wird als WM immer das Zeichen Punkt „." verwendet.

Die Bedeutung von K_1 = „:" und K_1 = „BM" wird im Abschnitt 3.2.42. erläutert.

2.2.1. $K_1 = 2$ Hilfsadressen

2.2.1.1. Leitadressen

Eingabebild: $2K_2$ $b_1b_2b_3b_4$ $a_1a_2a_3a_4$ WM

Es können max. 8 Oktalziffern, die zwei oktale Adressen $b = b_1b_2b_3b_4$ und $a = a_1a_2a_3a_4$ darstellen, entsprechend der im Abschnitt 1.1. dargestellten Triadenschreibweise eingegeben werden. Bei mehr als 8 Ziffern gehen die zuerst eingegebenen verloren.

Die Eingabe erfolgt ohne Konvertierung. Führende Nullen brauchen nicht gegeben zu werden, z. B. bei b = 0.

Die 8 Oktalziffern werden auf festen Speicherplätzen in den Oktalstellen t_3 bis t_{10} gespeichert (Abb. 4).

t_{10}	t_9	t_8	t_7	t_6	t_5	t_4	t_3	t_2	t_1	t_0
b_1	b_2	b_3	b_4	a_1	a_2	a_3	a_4	0	0	0

Abb. 4

In Abhängigkeit vom 2. Kennzeichen werden die verschiedenen Leitadressen eingestellt. Die Leitadressen werden erst durch die Eingabe entsprechender neuer Leitadressen zerstört.

1. Leitadresse

Eingabebild: 21 $b_1b_2b_3b_4$ $a_1a_2a_3a_4$ WM

Mit Hilfe der 1. Leitadresse wird der Speicherbefehl für die einzugebenden Worte in einer festen Zelle aufgebaut. Außerdem dient sie zur Adressenänderung 1. Art (s. K_1 = 0,1,4).

Die Abspeicherung der einzugebenden Worte erfolgt fortlaufend in natürlicher Reihenfolge ab der in der 1. LA angegebenen Adresse $a = a_1a_2a_3a_4$. Dabei wird nach jeder erfolgten Eingabe die Adresse a des Speicherbefehls um eine Einheit erhöht.

Beispiel 3: 21 2000.

Die Abspeicherung der nachfolgend einzugebenden Worte erfolgt ab Zelle 2000.

21 2000 3000.

Die Abspeicherung der einzulesenden Worte beginnt ab Speicherplatz 3000.

21 4100 4100.

Die Abspeicherung der einzugebenden Worte erfolgt ab Zelle 4100.

2. bis 9. Leitadresse

Eingabebilder:

22 $b_1b_2b_3b_4$ $a_1a_2a_3a_4$ WM 2. LA
23 $b_1b_2b_3b_4$ $a_1a_2a_3a_4$ WM 3. LA
.
.
.
29 $b_1b_2b_3b_4$ $a_1a_2a_3a_4$ WM 9. LA

Die 2. bis 9. Leitadresse haben keinen Einfluß auf die laufende Speicherung der Informationen. Sie dienen ausschließlich der Adressenänderung 1. Art (s. K_1 = 0,1,4).

Bemerkung: Mit Hilfe der Adresse b = $b_1b_2b_3b_4$ kann bei einem abzuspeichernden Maschinenwort eine Adressenänderung mit den vorderen 4 Triaden t_7 bis t_{10} durchgeführt werden.

2.2.1.2. Relativadresse

Eingabebild: 20 $a_1a_2a_3a_4$ WM

Bei der Relativadresse ist nur die Eingabe der Adresse a sinnvoll. Mit Hilfe dieser Adresse kann die Abspeicherung in natürlicher Reihenfolge unterbrochen werden. Sie bewirkt einen Neuaufbau des Speicherbefehls. Die Abspeicherung in natürlicher Reihenfolge wird ab der Adresse fortgesetzt, die sich aus der Addition der in der 1. LA angegebenen Adresse a und der Relativadresse ergibt. Diese Adressenaddition erfolgt oktal modulo 10 000. Die 1. LA wird dabei nicht zerstört. Die Relativadresse steht danach nicht mehr zur Verfügung.

Beispiel 4: 21 4100. 1. LA
20 20.
Die Abspeicherung der nachfolgenden Worte erfolgt ab Zelle 4120.
21 2000 2000.
20 5.
Die Abspeicherung der nachfolgenden Worte erfolgt ab Zelle 2005.

2.2.2. K_1 = 0 abzuspeicherndes Maschinenwort

Eingabebild: $0K_2$ $t_{10}t_9t_8t_7t_6t_5t_4t_3t_2t_1t_0$ WM

Es können max. 11 Oktalziffern entsprechend der Triadenschreibweise eingegeben werden. Bei mehr als 11 Ziffern gehen die zuerst eingegebenen verloren.

Die Eingabe erfolgt ohne Konvertierung. Führende Nullen brauchen nicht gegeben zu werden.

Ist eine Null auf einen Speicherplatz einzuschreiben, kann die kürzere Darstellung „0WM" (K_1 = 0, K_2 = WM) verwendet werden.

Mit Hilfe des ersten Kennzeichens K_1 = 0 können z. B. Maschinenbefehle[3]) und alphanumerische Zeichen, die entsprechend Tabelle 1 verschlüsselt sind, eingegeben werden.

In Abhängigkeit vom zweiten Kennzeichen K_2 wird vor der Abspeicherung eventuell eine Adressenänderung 1. Art durchgeführt. Diese Adressenänderung bezieht sich auf die Oktalstellen t_3 bis t_{10}. Damit können z. B. bei Maschinenbefehlen auch die ersten 4 Oktalstellen einer Adressenänderung unterzogen werden.

[3]) siehe Heft 2 „Anleitung zur Programmierung in der Maschinensprache"

K_2 kann die Werte 0 bis 9 mit folgender Bedeutung annehmen (vgl. $K_1 = 2$):

$K_2 = 0$ Das eingegebene Wort wird ohne Adressenänderung abgespeichert.

$K_2 = 1$ Vor der Abspeicherung des Wortes wird zu den Oktalstellen t_3 bis t_{10} die 1. LA addiert.

$K_2 = 2$ Vor der Abspeicherung des Wortes wird zu den Oktalstellen t_3 bis t_{10} die 2. LA addiert.

.
.
.

$K_2 = 9$ Vor der Abspeicherung des Wortes wird zu den Oktalstellen t_3 bis t_{10} die 9. LA addiert.

Beispiel 5:

21 2000 2000.	1. LA
23 4000.	3. LA
24 3000 0000.	4. LA

00 372 4570 421.

Das angegebene Maschinenwort wird ohne Adressenänderung abgespeichert.

01 13 0100 211.

Es wird das Maschinenwort 2013 2100 211 abgespeichert.

03 025.

Es wird das Wort 4000 025 abgespeichert.

04.

Es wird das Wort 3000 0000 000 abgespeichert.

2.2.3. $K_1 = 1$ Sofort-Maschinenbefehl[4])

Eingabebild: $1K_2\ b_1b_2b_3b_4\ a_1a_2a_3a_4$ BAZ WM

Es können max. 11 Oktalziffern entsprechend der Triadenschreibweise eingegeben werden. Bei mehr als 11 Ziffern gehen die zuerst eingegebenen verloren. Die Eingabe erfolgt ohne Konvertierung. Führende Nullen brauchen nicht gegeben zu werden (z. B. b = 0). Das zweite Kennzeichen K_2 hat dieselbe Bedeutung wie bei $K_1 = 0$.

Ein Sofortbefehl wird in einer festen Zelle gespeichert und anschließend angesprungen, d. h., er wird sofort abgearbeitet. Diese Befehle werden u. a. zum Anspringen von Maschinenprogrammen (Sprungbefehl) und zum Lesen von einzelnen Zellen (Lesebefehl) verwendet. Wenn der Sofortbefehl kein Sprungbefehl ist, geht der Automat nach Ausführung des Befehls in den Zustand „unbedingter Stop". Durch Drücken der Starttaste „ST" wird derselbe Befehl mit der um eine Einheit erhöhten Adresse durchgeführt. Danach geht der Automat wieder in „unbedingten Stop" usw. Eine Rückkehr zur Eingabe eines beliebigen ersten Kennzeichens ist durch die Schalterkombination „H G R" möglich.

Beispiel 6:

21 3000.	1. LA
22 3200.	2. LA

10 2000 000.

Sprung zur Adresse 2000 in ein Maschinenprogramm.

11.

Sprung zur Adresse 3000 in ein Maschinenprogramm.

---

[4]) Wenn man sich bei der Programmierung auf die interpretierende Arbeitsweise beschränkt, kann dieser Abschnitt überlesen werden. Die Darstellung der Maschinenbefehle ist in Heft 2 angegeben.

12 5 211.
Der Inhalt der Zelle 3205 wird in den Maschinenakkumulator gebracht.

Bemerkung: Die Darstellung „1WM" ($K_1 = 1$, $K_2 = WM$) hat dieselbe Wirkung wie „0WM" (s. $K_1 = 0$).

2.2.4. $K_1 = 4$ abzuspeichernder Externbefehl

Eingabebild: $4K_2\ S_1S_2\ I\ a_1a_2a_3a_4\ WM\ (p_3p_2p_1)$
Operationszusatz

Der Operationsteil des Externbefehls besteht aus den beiden Oktalziffern S_1 und S_2. Die Stelle I trägt die Nummer des jeweiligen Indexregisters. Die Grundadresse wird durch die vier Oktalziffern $a_1a_2a_3a_4$ angegeben. Führende Nullen bei der Adresse dürfen nicht weggelassen werden.

Bei einer Gruppe von Externbefehlen ist nach WM die Angabe eines Operationszusatzes notwendig. Dieser kann max. 3 Parameter p_1, p_2 und p_3 umfassen. Die Angaben erfolgen dabei dezimal. Das zweite Kennzeichen K_2 hat dieselbe Bedeutung wie bei $K_1 = 0$ bzw. $K_1 = 1$. Es gibt an, mit welcher Leitadresse eventuell vor der Abspeicherung eine Adressenänderung 1. Art erfolgt. Diese Adressenänderung wird bei den Externbefehlen so durchgeführt, daß zur Grundadresse a die in der entsprechenden Leitadresse angegebene Adresse a oktal modulo 10 000 addiert wird.

Bei der Eingabe wird der Externbefehl in die in Abb. 2 angegebene interne Darstellung überführt. Dabei erfolgt auch eine Umschlüsselung des Operationscodes S_1S_2.

Die Externbefehle werden wegen ihrer zentralen Stellung im System „FIPS I" gesondert behandelt (s. Kapitel 3).

Bemerkung: Die Darstellung „4WM" ($K_1 = 4$; $K_2 = WM$) hat dieselbe Wirkung wie „0WM" bzw. „1WM": Abspeichern einer Null (s. $K_1 = 0$).

Beispiel:

21 3000 2000.		1. LA
22	2500.	2. LA

41 71 5 0005.
Es wird der Externbefehl 71 5 2005 in die Interndarstellung überführt und abgespeichert.
42 40 2 0000.
Es wird der Externbefehl 40 2 2500 in die interne Darstellung überführt und abgespeichert.
40 30 1 0100.
Es wird der Externbefehl 30 1 0100 in die Interndarstellung überführt und abgespeichert.

2.2.5. $K_1 = 5$ Sprungbefehl extern

Eingabebild: $5\ a_1a_2a_3a_4\ WM\ b_1b_2b_3b_4\ WM$

Nach dem ersten Kennzeichen $K_1 = 5$ sind zwei oktale Adressen a und b, jeweils durch WM abgeschlossen, einzugeben. Führende Nullen können bei der Angabe der Adressen unterdrückt werden.

Mit Hilfe dieses Sprungbefehls kann ein Programm, das sich auf die interpretierende Abarbeitung von Extern- bzw. Maschinenbefehlen bezieht, angesprungen werden. Die Ansprungadresse wird durch die Adresse $a = a_1a_2a_3a_4$ bestimmt.

Mit Hilfe der zweiten Adresse $b = b_1b_2b_3b_4$ kann für Testzwecke nach einem Teil der interpretierend abzuarbeitenden Befehle eine Programmunterbrechung er-

reicht werden. Dabei wird das laufende Programm nach der Abarbeitung des in der Zelle b gespeicherten Befehls gestoppt. Danach können über Hilfsprogramme die Inhalte von Zellen, z. B. von R und I_n ($n = 1, \ldots, 7$) ausgeschrieben werden. Anschließend ist eine Programmfortsetzung möglich (s. Abschnitt 6.2.). Wird keine Unterbrechung des laufenden Programms gewünscht, ist die Adresse b zu unterdrücken ($b = 0$).

Beispiel 8: 5 2000. 2126.
Sprung zur Adresse 3000 in ein interpretierend arbeitendes Programm.
Unterbrechung des Programms nach Abarbeitung des Befehls auf 2126.
5 3000. .
Sprung zur Adresse 2000 in ein interpretierend arbeitendes Programm.

2.2.6. $K_1 = 3$ Dezimalzahl

Eingabebild: 3 (–) $d_9d_8d_7d_6d_5d_4d_3d_2d_1d_0$ WM

Nach dem Kennzeichen $K_1 = 3$ folgt eine Dezimalzahl in üblicher Schreibweise. Bei der Eingabe wird das Komma überlesen. Die Zahl wird als ganz interpretiert. Die Kommastellung der einzugebenden Zahl ist vom Programmierer zu berücksichtigen bzw. bei der Erstellung der Daten zu beachten.

Ein positives Vorzeichen ist zu unterdrücken. Bei einer negativen Zahl kann das Vorzeichen an beliebiger Stelle vor dem Schlußzeichen WM stehen.

Es können ganze Zahlen aus dem Bereich
$|D| \leq 2^{31}-1 = 2\ 147\ 483\ 647$
eingegeben werden. Bei Überschreiten des Zahlenbereiches wird die Zahl verfälscht. Eine Null kann durch die Darstellung „3WM" ($K_1 = 3$, $d_i = 0$, $i = 0, \ldots, 9$) eingegeben werden.

Die Dezimalzahl wird bei der Eingabe in die im Abschnitt 1.2. beschriebene Interndarstellung konvertiert.

Beispiel 9: Die Zahlen 12,25; 0,35 und –13,3 sind mit gleicher Stellenanzahl (2 Stellen nach dem Komma) einzugeben:

3 12,25.
3 0,35.
3 —13,30. oder 3 13,30—. oder auch 3 13—30.

Die dritte Zahl wurde aufgefüllt. Die Kommas werden überlesen, so daß die Zahlen 1225; 35 und —1330 gespeichert werden.

Zusammenfassendes Beispiel:

Es liegt ein Hauptprogramm vor, das relativ bezüglich der 1., 2. und 4. LA programmiert ist.

Ab 2. LA steht ein Unterprogramm, ab 4. LA ein Zahlenfeld. Die Eingabe der Programme, Zahlen und Hilfsadressen erfolgt in der Reihenfolge (1. LA = 3000, 2. LA = 4000, 4. LA = 4300):

1. 21 3000. 1. LA des Hauptprogramms
2. 22 4000. 2. LA
3. 24 4300. 4. LA
4. Eingabe – Hauptprogramm
5. 21 4000. 1. LA des UP = 2. LA des Hauptprogramms
6. Eingabe – Unterprogramm
7. 21 4300. 1. LA des Zahlenfeldes = 4. LA des Hauptprogramms
8. Eingabe – Zahlenfeld

2.2.7. $K_1 = 8$ Dezimalzahlen im Satz

Durch das 1. Kennzeichen $K_1 = 8$ wird ein Programm ausgewählt, das die hintereinanderfolgende Eingabe von Dezimalzahlen ohne Angabe eines 1. Kennzeichens ermöglicht.
Es werden Dezimalzahlen der gleichen Darstellung wie bei $K_1 = 3$ eingelesen. Zur Eingabe einer Null genügt die Angabe der Wortmarke.
Die Anfangsadresse der fortlaufenden Speicherung wird durch die Adresse a der 1. LA bestimmt. Sie ist vor dem ersten Kennzeichen $K_1 = 8$ zu geben. Die einzugebenden Zahlen können satzweise durch das Zeichen „Satzmarke/WRZL" in der 8-Kanal-Variante und durch das Zeichen „:" in der 5-Kanal-Variante zusammengefaßt werden. Diese Zeichen schließen die Funktion der Wortmarke nicht ein.
Ein Satz von Dezimalzahlen kann in der 8-Kanal-Version durch die Zeichen „Irrung Satz" bzw. Apostroph „'" und in der 5-Kanal-Version durch das Zeichen „H" gelöscht werden. Das Irrungszeichen kann an einer beliebigen Stelle des Satzes, aber vor dem entsprechenden Satzschlußzeichen, gegeben werden.
Ein besonderes Lochbandschlußzeichen existiert nicht. Die Rückkehr zur Eingabe eines beliebigen ersten Kennzeichens ist durch die Schalterkombination H G R möglich.

Beispiel 10: Die Zahlen 18 543; 15 667; −18 241; −1226; −15; 220 sind ab 3000 zu speichern.
Eingabe der 1. LA 21 3000.
Eingabe des ersten Kennzeichens $K_1 = 8$
Eingabe der Zahlen (hier zu zwei Sätzen zusammengegefaßt):

18 543.
15 667.
−18 241.
Satzschlußzeichen („WRZL" bzw. „:")
−1 236.
−15.
220.
Irrungszeichen („'" bzw. „H")
−1 226.
−15.
220.

Die Zahlen werden auf den Speicherplätzen 3000 bis 3005 gespeichert. Auf das Satzschlußzeichen des letzten Satzes kann verzichtet werden.

2.2.8. $K_1 = WM$ Wortmarke

Das erste Kennzeichen $K_1 = WM$ (Punkt „." bzw. „Tab" bei 8-Kanal, Punkt „." bei 5-Kanal) führt zum Löschen des durch den Abspeicherbefehl bestimmten Speicherplatzes. Es hat dieselbe Wirkung wie zum Beispiel die Darstellung „0WM".
($K_1 = 0$, $K_2 = WM$; s. Abschnitt 2.2.2.)

2.2.9. $K_1 = /$ Lochbandschlußzeichen

Mit Hilfe dieses Kennzeichens kann der Lochbandtransport bei der Eingabe mit Kennzeichen gestoppt werden. Beim Lesen dieses Zeichens geht der Automat, falls der Selektor „BS" eingerastet ist, in den Zustand „bedingter Stop". Die Eingabe wird mit der Eingabe eines ersten Kennzeichens fortgesetzt, wenn der Schalter „BS" gelöst oder die Taste „ST" gedrückt wird oder wenn der Schalter „BS" nicht eingerastet war.

2.2.10. $K_1 = ZR$ Zwischenraum

„ZR" als erstes Kennzeichen ist wirkungslos. Es wird überlesen. Es kann zur Gliederung des Lochbandes benutzt werden.

2.2.11. $K_1 = B$, $K_1 = D$ Programme mit B- und D-Taste

Mit Hilfe dieser Kennzeichen werden Bedienungs- und Kontrollprogramme angesprungen. Die Beschreibung dieser Programme erfolgt in dem Kapitel 6.

3. Externbefehle

3.1. Indexregister und Indexmodifikation

Die Indexregister haben eine Wortlänge von 33 Bits. Sie können z. B. zur Zwischenspeicherung von Adressen und Zahlen verwendet werden. Eine Gruppe von Externbefehlen ermöglicht die Verknüpfung des Inhaltes einer Speicherzelle mit dem Inhalt eines Indexregisters I_n ($n = 1, \ldots, 7$).

Darüber hinaus kann mit Hilfe der Indexregister bei einer Anzahl von Externbefehlen eine Indexmodifikation durchgeführt werden. Dabei wird zur Grundadresse a des Externbefehles vor dessen Ausführung die in den Oktalstellen t_3 bis t_6 des Indexregisters I_n gespeicherte Adresse addiert. Diese Adressenaddition wird oktal modulo 10 000 ausgeführt.

Das Indexregister I_n ist durch seine Nummer n ($n = 1, \ldots, 7$) in der Spalte I des Externbefehles anzugeben. n gleich Null besagt, daß keine Modifikation durchgeführt wird.

Bemerkung: Im weiteren Verlauf wird die modifizierte Grundadresse a eines Externbefehles mit a* bezeichnet. Die Oktalstellen t_3 bis t_6 eines Maschinenwortes werden auch als Adressenstellen bezeichnet.

Beispiel 11: Indexmodifikation

Grundadresse a	2000	
Triadendarstellung von $\langle I_1 \rangle$	7777	0005 011
Indexmodifikation mit $\langle I_1 \rangle$	2000	
	5	
modifizierte Adresse a*	2005	
Grundadresse a	2000	
Triadendarstellung von $\langle I_2 \rangle$	7777	7776 000
Indexmodifikation mit $\langle I_2 \rangle$	2000	
	7776	
modifizierte Adresse a*	(1) 1776	

Die Addition der Adresse 7776 entspricht infolge der Modulo-Rechnung einer Subtraktion von 2 Einheiten.

3.2. Erläuterung der Externbefehle

Zur Erläuterung der Wirkungsweise der Externbefehle werden einfache Beispiele angeführt. Dabei wurde darauf geachtet, daß kein Befehl verwendet wird, der noch nicht erklärt worden ist.

3.2.1. $S_1S_2 = 01$ Lesebefehl

Eingabebild: $4K_2$ 1 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt der Zelle a* wird in das Register R gebracht.
Der Inhalt von a* wird nicht verändert.
$\langle R \rangle := \langle a^* \rangle$

3.2.2. $S_1S_2 = 03$ Transportbefehl

Eingabebild: $4K_2$ 3 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt des Registers R wird in die Zelle a* transportiert.
Der alte Inhalt von a* wird dabei überschrieben.
Der Inhalt von R bleibt erhalten.
$\langle a^* \rangle := \langle R \rangle$

Beispiel 12: Die Inhalte der Zellen 7741 und 7742 sind zu vertauschen.

Die Zelle 7743 kann als Arbeitszelle verwendet werden. Die Befehlsfolge ist ab 2000 gespeichert.

laufende Adressierung	Befehl mit Eingabekennzeichen	Bemerkungen
2000	40 1 0 7741.	⟨R⟩ : = ⟨7741⟩
2001	40 3 0 7743.	⟨7743⟩ : = ⟨R⟩
2002	40 1 0 7742.	⟨R⟩ : = ⟨7742⟩
2003	40 3 0 7741.	⟨7741⟩ : = ⟨R⟩
2004	40 1 0 7743.	⟨R⟩ : = ⟨7743⟩
2005	40 3 0 7742.	⟨7742⟩ : = ⟨R⟩

3.2.3. S_1S_2 = 02 Lesebefehl, negativ

Eingabebild: $4K_2$ 2 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt des Speicherplatzes a* wird mit umgekehrtem Vorzeichen in das Register R gebracht. Der Inhalt der Zelle a* bleibt erhalten.

⟨R⟩ : = – ⟨a*⟩

Bemerkung: Ist bei der Zelle a* nur das Vorzeichenbit z_0 besetzt, geht der Automat in den Zustand „Überlauf".

Beispiel 13: Die in der Zelle 3000 gespeicherte Zahl D ist in die Zelle 7741 umzuspeichern. In der Zelle 7742 ist die Zahl –D bereitzustellen. Die Befehle sind ab 6000 gespeichert.

6000 40 1 0 3000. ⟨R⟩ : = D
6001 40 3 0 7741. ⟨7741⟩ = D
6002 40 2 0 3000. ⟨R⟩ : = –D
6003 40 3 0 7742. ⟨7742⟩ = –D

3.2.4. S_1S_2 = 04 Löschbefehl

Eingabebild: $4K_2$ 4 I $a_1a_2a_3a_4$ WM

Wirkung: In die Speicherzelle a* und in das Register R wird jeweils eine Null eingetragen.

⟨R⟩ : = ⟨a*⟩ : = 0

3.2.5. S_1S_2 = 71 Indexregister – Lesebefehl

Eingabebild: $4K_2$ 71 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt der Zelle a wird in das Indexregister I_n (n = 1, . . ., 7) übertragen. Der Inhalt von a wird nicht verändert.

$\langle I_n \rangle$: = ⟨a⟩

Beispiel 14: Die Inhalte der Indexregister I_1 und I_2 sind zu vertauschen. Zur Zwischenspeicherung wird I_3 benutzt. Die Befehle sind ab 2000 gespeichert (vgl. Beispiel 12).

2000 40 71 3 7741. $\langle I_3 \rangle$: = $\langle I_1 \rangle$
2001 40 71 1 7742. $\langle I_1 \rangle$: = $\langle I_2 \rangle$
2002 40 71 2 7743. $\langle I_2 \rangle$: = $\langle I_3 \rangle$

Beispiel 15: Die Zahlen D_1, D_2 und D_3 sind in hintereinanderfolgenden Zellen gespeichert. Die Adresse von D_1 steht in den Adressenstellen der Zelle 7000. Die Zahlen sind umzuspeichern.

Die zugehörige Anfangsadresse ist in den Adressenstellen der Zelle 7001 gespeichert. Die Befehle stehen ab 3000.

3000	40	71	1	7000.	$\langle I_1 \rangle := \langle 7000 \rangle$
3001	40	71	2	7001.	$\langle I_2 \rangle := \langle 7001 \rangle$
3002	40	1	1	0000.	$\langle R \rangle := D_1$
3003	40	3	2	0000.	
3004	40	1	1	0001.	$\langle R \rangle := D_2$
3005	40	3	2	0001.	
3006	40	1	1	0002.	$\langle R \rangle := D_3$
3007	40	3	2	0002.	

3.2.6. $S_1S_2 = 73$ Indexregister-Transportbefehl

Eingabebild: $4K_2$ 73 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt des angegebenen Indexregisters I_n ($n = 1, \ldots, 7$) wird in die Zelle a gebracht. Der alte Inhalt von a wird dabei überschrieben.

Der Inhalt des Indexregisters bleibt erhalten.

$\langle a \rangle := \langle I_n \rangle$

Beispiel 16: Die Inhalte der Zellen 2000 und 2001 sind zu vertauschen. Zur Zwischenspeicherung ist I_7 zu verwenden. Der alte Inhalt von I_7 ist zu erhalten. Die Befehle stehen ab 3000.

3000	40	73	7	7000.	Zwischensp. von $\langle I_7 \rangle$
3001	40	71	7	2000.	$\langle I_7 \rangle := \langle 2000 \rangle$
3002	40	1	0	2001.	$\langle R \rangle := \langle 2001 \rangle$
3003	40	3	0	2000.	
3004	40	73	7	2001.	$\langle 2001 \rangle := \langle I_7 \rangle$
3005	40	71	7	7000.	$\langle I_7 \rangle := \langle 7000 \rangle$

3.2.7. $S_1S_2 = 72$ Indexregister-Lesebefehl, negativ

Eingabebild: $4K_2$ 72 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt der Zelle a wird mit umgekehrtem Vorzeichen in das Indexregister I_n gebracht. Der Inhalt der Zelle a bleibt erhalten.

$\langle I_n \rangle := -\langle a \rangle$

Bemerkung: Ist bei der Zelle a nur das Vorzeichenbit z_0 besetzt, kommt es zum Überlauf.

3.2.8. $S_1S_2 = 74$ Indexregister-Löschbefehl

Eingabebild: $4K_2$ 74 I $a_1a_2a_3a_4$ WM

Wirkung: In die Zelle a und in das Indexregister I_n wird jeweils eine Null eingetragen.

$\langle a \rangle := \langle I_n \rangle := 0$

3.2.9. $S_1S_2 = 05$ Addition

Eingabebild: $4K_2$ 5 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt der Zelle a^* wird zum Inhalt des Registers R addiert. Das Ergebnis steht in R. Der Inhalt der Zelle a^* wird nicht verändert. Bei Kapazitätsüberschreitung kommt es zum Überlauf.

$\langle R \rangle := \langle R \rangle + \langle a^* \rangle$

Beispiel 17: Es ist die Summe der drei Zahlen D_1, D_2 und D_3 zu bilden und in I_1 zu speichern. Die Zahlen stehen in hintereinanderfolgenden Zellen. Die Anfangsadresse ist in den Adressenstellen von I_2 gespeichert.

Die Befehlsfolge ist ab 2000 untergebracht.

```
2000 40 1 2 0000.   ⟨R⟩ : = D1
2001 40 5 2 0001.   ⟨R⟩   = D1 + D2
2002 40 5 2 0002.   ⟨R⟩   = D1 + D2 + D3
2003 40 3 0 7741.   ⟨I1⟩ : = ⟨R⟩
```

3.2.10. $S_1S_2 = 06$ Subtraktion

Eingabebild: $4K_2$ 6 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt der Zelle a^* wird vom Inhalt des Registers R subtrahiert. Das Ergebnis steht in R. Der Inhalt von a^* bleibt erhalten. Bei Kapazitätsüberschreitung kommt es zum Überlauf.

$\langle R\rangle := \langle R\rangle - \langle a^*\rangle$

Beispiel 18: Die Zahlen D_1, D_2, D_3 sind in hintereinanderfolgenden Zellen gespeichert. Die Anfangsadresse steht in den Adressenstellen von I_6. Es sind die Ausdrücke $S = D_1 + D_2 - D_3$ und $-S$ zu bilden und in I_1 bzw. I_2 zu speichern.

Die Befehle stehen ab 7000.

```
7000 40  1 6 0000.   ⟨R⟩ : = D1
7001 40  5 6 0001.
7002 40  6 6 0002.   ⟨R⟩   = D1 + D2 — D3
7003 40  3 0 7741.
7004 40 72 2 7741.   ⟨I2⟩ : = —⟨I1⟩
```

Beispiel 19: Die in der Zelle 7000 stehende Information ist zu negieren; in alle Bits, die mit 1 besetzt sind, ist eine Null einzutragen und umgekehrt.

Die Befehle sind ab 2000 gespeichert.

```
2000 40 2 0 7000.   ⟨R⟩ : = —⟨7000⟩
2001 40 6 0 7001.
2002 40 3 0 7000.
7001 00 1.
```

3.2.11. $S_1S_2 = 07$ Konjunktion, binär

Eingabebild: $4K_2$ 7 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt des Registers R und der Inhalt der Zelle a^* werden bitweise durch eine Konjunktion verknüpft. Das Ergebnis steht in R. Der Inhalt von a^* wird nicht verändert.

$\langle R\rangle := \langle R\rangle \wedge \langle a^*\rangle$

Beispiel 20: Von dem in der Zelle 2010 stehenden Maschinenwort sollen die Oktalstellen t_7 bis t_{10} ausgeblendet und in I_1 gespeichert werden. Die Oktalstellen t_0 bis t_6 sind ebenfalls auszublenden und in I_2 zu speichern. Als Konstante steht in 2011 die Maske M = 7777 0000 000 (Triadenschreibweise) zur Verfügung. In der Zelle 2012 ist nur das Bit z_{32} besetzt.

Die Befehlsfolge steht ab 2000.

```
2000 40 1 0 2010.   ⟨R⟩ : = ⟨2010⟩
2001 40 7 0 2011.   ⟨R⟩ : = ⟨R⟩ ∧ M
2002 40 3 0 7741.   ⟨I1⟩ : = ⟨R⟩
2003 40 2 0 2011.   ⟨R⟩ : = — M
2004 40 6 0 2012.
2005 40 7 0 2010.
2006 40 3 0 7742.
2011 00 7777 0000 000.
2012 00 1.
```

3.2.12. $S_1S_2 = 10$ Vergleichen

Eingabebild: $4K_2$ 10 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt der Zelle a* wird mit dem Inhalt des Registers R verglichen. Wenn die in a* gespeicherte Zahl größer als die im Register R gespeicherte ist, wird ein Vergleicher „VG2" gesetzt. Andernfalls wird ein Vergleicher „VG1" gesetzt.

Der Vergleichszustand kann durch entsprechende Befehle abgefragt werden.

3.2.13. $S_1S_2 = 75$ Indexregister-Addition

Eingabebild: $4K_2$ 75 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt der Zelle a wird zum Inhalt des Indexregisters I_n addiert. Das Ergebnis steht in I_n. Der Inhalt von a wird nicht verändert. Bei Kapazitätsüberschreitung kommt es zum Überlauf.

$\langle I_n \rangle := \langle I_n \rangle + \langle a \rangle$

Beispiel 21: Die in hintereinanderfolgenden Zellen gespeicherten Zahlen D_1 und D_2 sind umzuspeichern. Die Adresse von D_1 ist in den Adressenstellen von I_1 gespeichert. In den Adressenstellen von I_2 befindet sich die Differenz zwischen der Anfangsadresse der umgespeicherten und der der gespeicherten Werte.

Die Befehle sind ab 2000 untergebracht.

```
2000 40 75 2 7741.   ⟨I2⟩ := ⟨I2⟩ + ⟨I1⟩
2001 40  1 1 0000.
2002 40  3 2 0000.
2003 40  1 1 0001.   ⟨R⟩ := D2
2004 40  3 2 0001.
```

3.2.14. $S_1S_2 = 76$ Indexregister-Subtraktion

Eingabebild: $4K_2$ 76 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt der Zelle a wird vom Inhalt des Indexregisters I_n subtrahiert. Das Ergebnis steht in I_n. Der Inhalt der Zelle a bleibt erhalten. Bei Kapazitätsüberschreitung entsteht Überlauf.

$\langle I_n \rangle := \langle I_n \rangle - \langle a \rangle$

Beispiel 22: Die vier Zahlen D_1, D_2, D_3 und D_4 sind in den Zellen 3000 bis 3003 gespeichert. Es sollen die Werte $D_1 - D_2$, $D_3 - D_4$ und $D_1 - D_2 - D_3 + D_4$ gebildet und im Register R bzw. in I_1 und I_2 gespeichert werden.

Die Befehle sind ab 3004 gespeichert.

```
3004 40  1 0 3000.
3005 40  6 0 3001.   ⟨R⟩  = D1 — D2
3006 40 71 1 3002.
3007 40 76 1 3003.   ⟨I1⟩ = D3 — D4
3010 40  3 0 7742.   ⟨I2⟩ := ⟨R⟩ = D1 — D2
3011 40 76 2 7741.   ⟨I2⟩ = D1 — D2 — (D3 — D4)
```

3.2.15. $S_1S_2 = 77$ Indexregister-Konjunktion, binär

Eingabebild: $4K_2$ 77 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt des Indexregisters I_n und der Inhalt der Zelle a werden bitweise durch Konjunktion verknüpft. Das Ergebnis steht in I_n. Der Inhalt von a wird nicht verändert.

$\langle I_n \rangle := \langle I_n \rangle \wedge \langle a \rangle$

Beispiel 23: In der Zelle 7000 ist in den Oktalstellen t_7 bis t_{10} die Adresse $b = b_1b_2b_3b_4$ gespeichert. Die Zellen 7001 und 7002 sind folgendermaßen belegt:

$\langle 7001 \rangle = 7777\ 0000\ 000$ (Triadenschreibweise)
$\langle 7002 \rangle = \qquad 12\ 000$

Es sollen in den Adressenstellen von I_1 die Zahl $n = -10$ ($n_{okt} = -12$) und in den Oktalstellen t_7 bis t_{10} von I_1 die Adresse $b-1$ gespeichert werden.

Die Befehlsfolge steht ab 2000.

```
2000 40 71 1 7000.  ⟨I1⟩ : = ⟨7000⟩
2001 40 77 1 7001.  ⟨I1⟩   = b1b2b3b4 0000 000
2002 40 76 1 7002.
```

Beispiel 24: Die in den Zellen 7000 und 7002 gespeicherten Werte D_1 und D_2 sind bitweise durch Disjunktion zu verknüpfen:

$D_1 \vee D_2 = (D_1 \wedge \neg D_2) + D_2$

Das Ergebnis ist in I_6 zu speichern. Die Befehlsfolge steht ab 5000.

```
5000 40 72 6 7002.  ⟨I6⟩ : = —D2
5001 40 76 6 7001.
5002 40 77 6 7000.  ⟨I6⟩   = D1 ∧ ¬ D2
5003 40 75 6 7002.
7001 00 1.
```

3.2.16. $S_1S_2 = 63$ Indexregister-Vergleichen

Eingabebild: $4K_2$ 63 I $a_1a_2a_3a_4$ WM

Wirkung: Der Inhalt der Zelle a wird mit dem Inhalt des Indexregisters I_n verglichen.

Wenn die in der Zelle a gespeicherte Zahl größer als die im Indexregister I_n gespeicherte ist, wird der Vergleicher „VG2" gesetzt. Andernfalls wird der Vergleicher „VG1" gesetzt.

3.2.17. $S_1S_2 = 12$ Betragsbildung

Eingabebild: $4K_2$ 12 I $a_1a_2a_3a_4$ WM

Wirkung: Es wird der Betrag der in der Zelle a^* gespeicherten Zahl gebildet und ins Register R transportiert. Der Inhalt von a^* bleibt erhalten.

$\langle R \rangle := |\langle a^* \rangle|$

Beispiel 25: Die Zahlen D_1, D_2, D_3, D_4 sind in hintereinanderfolgenden Zellen gespeichert. Die Adresse von D_4 ist in den Adressenstellen von I_2 gespeichert.

Es ist der Ausdruck $S = D_1 + |D_2| + D_3 - |D_4|$ zu bilden und in I_1 zu speichern.

Die Befehlsfolge steht ab 2000.

```
2000 40 12 2 7776.  ⟨R⟩ : = |D2|
2001 40  5 2 7775.
2002 40  5 2 7777.  ⟨R⟩   = |D2| + D1 + D3
2003 40  3 0 7741.
2004 40 12 2 0000.  ⟨R⟩ : = |D4|
2005 40 76 1 7773.
```

3.2.18. $S_1S_2 = 13$ Verschiebung links

Eingabebild: $4K_2$ 13 I $a_1a_2a_3a_4$ WM v

Wirkung: Die in der Zelle a^* stehende Dezimalzahl wird um v Dezimalstellen ($v = 1, \ldots, 9$) links verschoben in das Register R übertragen (Multiplikation mit 10^v).

Bei Kapazitätsüberschreitung wird der Wert verfälscht. Eine Fehleranzeige ist nicht berücksichtigt. Der Inhalt von a^* bleibt erhalten. In der Eingabedarstellung ist der Parameter $p_1 = v$ nach WM anzugeben.

$\langle R \rangle := \langle a^* \rangle \times 10^v$

Bemerkung: Das Eingabebild dieses Externbefehles umfaßt einschließlich WM immer 11 Zeichen.

Beispiel 26: Die Zahlen $D_1 = 210$, $D_2 = 3501$ und $D_3 = 990$ sind in drei hintereinanderfolgenden Zellen gespeichert. Die zugehörige Anfangsadresse ist in den Adressenstellen von I_1 gespeichert. Die drei Zahlen sollen gemeinsam im Indexregister I_3 gespeichert werden.

Die Befehlsfolge ist ab 5000 untergebracht.

5000	40 13 1 0000. 4	$\langle R \rangle = 210\ 0000$
5001	40 5 1 0001.	$\langle R \rangle = 210\ 3501$
5002	40 13 0 7773. 3	$\langle R \rangle = 210\ 3501\ 000$
5003	40 5 1 0002.	$\langle R \rangle = 210\ 3501\ 990$
5004	40 3 0 7743.	

Beispiel 27: Die drei in hintereinanderfolgenden Zellen gespeicherten Zahlen $D_1 = 21{,}5$; $D_2 = 5{,}37$ und $D_3 = 16{,}78$ sind mit 2 Stellen nach dem Komma aufzusummieren. Die Kommastellung ist nur gedacht. Gespeichert sind ganze Zahlen. Die Adresse von D_1 ist in den Adressenstellen von I_6 gespeichert.

Die Befehlsfolge steht ab 2000.

2000	40 13 6 0000. 1	$\langle R \rangle = 2150$
2001	40 5 6 0001.	
2002	40 5 6 0002.	

3.2.19. $S_1S_2 = 14$ Verschiebung rechts

Eingabebild: $4K_2$ 14 I $a_1a_2a_3a_4$ WM v

Wirkung: Die in der Zelle a^* stehende Dezimalzahl wird um v Dezimalstellen ($v = 1, \ldots, 9$) rechts verschoben in das Register R übertragen (Multiplikation mit 10^{-v}). Die Verschiebung erfolgt ohne Rundung. Der Inhalt von a^* bleibt erhalten. Der Parameter $p_1 = v$ wird im Eingabebild nach WM angegeben.

$\langle R \rangle := \langle a^* \rangle \times 10^{-v}$

Bemerkung: Das Eingabebild dieses Externbefehls umfaßt einschließlich WM immer 11 Zeichen.

Beispiel 28: Mit Hilfe der Verschiebebefehle kann eine dezimale Konjunktion realisiert werden. In der Zelle 2000 ist die Dezimalzahl D = —593 469 211 gespeichert. Es soll die in den Dezimalstellen d_3 bis d_5 stehende positive Zahl z = 469 in I_2 gespeichert werden.

Die Befehlsfolge ist ab 3000 gespeichert.

2000	3 —593 469 211.	
3000	40 14 0 2000. 3	$\langle R \rangle = -593\ 469$
3001	40 3 0 7743.	zwischenspeichern
3002	40 14 0 7773. 3	$\langle R \rangle = -593$
3003	40 13 0 7773. 3	$\langle R \rangle = -593\ 000$
3004	40 6 0 7743.	$\langle R \rangle = 469$
3005	40 3 0 7742.	

3.2.20. $S_1S_2 = 15$ Multiplikation

Eingabebild: $4K_2$ 15 I $a_1a_2a_3a_4$ WM v

Wirkung: Die im Register R stehende Dezimalzahl (Multiplikand) wird mit der in der Zelle a^* gespeicherten Dezimalzahl (Multiplikator) vorzeichenrichtig multipliziert. Anschließend an diese Multiplikation kann im gleichen Befehl eine Rechtsverschiebung um v Dezimalstellen ($v = 0, \ldots, 9$) programmiert werden.

Der Parameter $p_1 = v$ ist entsprechend dem Eingabebild nach WM einzugeben.

Die Verschiebung wird mit Runden durchgeführt. Bei Kapazitätsüberschreitung in der einfachen Wortlänge bezüglich des Produktes $\langle R \rangle \times \langle a^* \rangle \times 10^{-v}$ geht der Automat in Überlauf. Das Produkt steht nach Beendigung der Operation im Register R.

Der Inhalt von a^* bleibt erhalten.

$\langle R \rangle := \langle R \rangle \times \langle a^* \rangle \times 10^{-v}$

Bemerkung: Die gedachte Kommastellung des Ergebnisses ist durch die Verschiebung zu realisieren. Das Eingabebild dieses Externbefehls umfaßt einschließlich WM immer 11 Zeichen.

Beispiel 29: Die Zahlen $D_1 = 185{,}41$; $D_2 = 15{,}667$; $D_3 = 182{,}41$ und $D_4 = 151{,}213$ sind in hintereinanderfolgenden Zellen gespeichert.

Die Adresse von D_4 steht in den Adressenstellen von I_4. In der Zelle 3000 ist eine 1 gespeichert. Es ist die Summe $S = D_1D_2 + D_3D_4$ zu bilden und in I_1 zu speichern. Die Summanden sind mit 2 Stellen nach dem Komma zu berechnen, die Summe S mit einer Stelle.

Diese Stelle ist zu runden.

Die Befehlsfolge steht ab 3001.

3000	3	1.						
3001	40	1	4	7775.		$\langle R \rangle =$	185,41
3002	40	15	4	7776.	3	$\langle R \rangle =$	2 904,82
3003	40	3	0	7741.			
3004	40	1	4	7777.		$\langle R \rangle =$	182,41
3005	40	15	4	0000.	3	$\langle R \rangle =$	27 582,76
3006	40	5	0	7741.		$\langle R \rangle =$	30 487,58
3007	40	15	0	3000.	1		
3010	40	3	0	7741.		$\langle I_1 \rangle =$	30 487,6

3.2.21. $S_1S_2 = 16$ Division

Eingabebild: $4K_2$ 16 I $a_1a_2a_3a_4$ WM v

Wirkung: Die im Register R stehende Dezimalzahl (Dividend) wird durch die in der Zelle a^* stehende Dezimalzahl (Divisor) vorzeichenrichtig dividiert.

Vor Ausführung der Division ist es möglich, den Dividenden um v Dezimalstellen ($v = 0, \ldots, 9$) nach links zu verschieben. Der Parameter $p_1 = v$ ist entsprechend dem Eingabebild nach WM zu geben. Bei der Verschiebung kann die Kapazität der einfachen Wortlänge überschritten werden, wenn das Produkt $\langle R \rangle \times 10^v$ durch den Divisor auf einfache Wortlänge reduziert wird.

Bei Kapazitätsüberschreitung in der einfachen Wortlänge bezüglich des Quotienten $\langle R \rangle \times 10^v / \langle a^* \rangle$ geht

der Automat in Überlauf. Der Quotient steht nach Beendigung der Operation im Register R. Die letzte Quotientenstelle ist gerundet.

Der Inhalt von a* wird nicht zerstört.

$\langle R \rangle := \langle R \rangle \times 10^{v} / \langle a^* \rangle$

Bemerkung: Die Genauigkeit des Ergebnisses ist durch die Verschiebung zu realisieren. Das Eingabebild dieses Externbefehls umfaßt einschließlich WM immer 11 Zeichen.

Beispiel 30: In drei hintereinanderfolgenden Zellen stehen die Zahlen D_1, D_2 und D_3. Die Zahlen D_2 und D_3 besitzen 2 Stellen nach dem Komma, die Zahl D_1 eine Stelle. Die Adresse von D_2 steht in den Adressenstellen von I_2. Es ist der Quotient $Q = D_1/D_2$ mit 2 Stellen nach dem Komma zu berechnen und in 7000 zu speichern. Anschließend ist der Ausdruck $P = 100 \times Q/D_3$ mit einer Stelle nach dem Komma zu bilden und in 7001 zu speichern.

Die Befehlsfolge ist ab 5000 untergebracht.

```
5000 40  1 2 7777.        ⟨R⟩ : = D1
5001 40 16 2 0000. 3
5002 40  3 0 7000.       ⟨7000⟩ = Q
5003 40 16 2 0001. 3
5004 40  3 0 7001.       ⟨7001⟩ = P
```

3.2.22. $S_1S_2 = 17$ Konjunktion dezimal

Eingabebild: $4K_2$ 17 | $a_1a_2a_3a_4$ WM p_2p_1

Wirkung: Durch diesen Befehl kann eine zusammenhängende Folge von Ziffern einer Dezimalzahl $D = (-)d_9d_8d_7d_6d_5d_4d_3d_2d_1d_0$, die in der Zelle a* gespeichert ist, ausgeblendet werden.

Zur Ausführung der Operation ist die Angabe zweier Parameter p_1 ($p_1 = 0, \ldots, 9$) und p_2 ($p_2 = 1, \ldots, 9$) in der Operationsergänzung notwendig. Der Parameter p_1 bestimmt die kleinste Stelle der auszublendenden Ziffernfolge. Der Parameter p_2 gibt die Anzahl der auszublendenden Ziffern an.

Im Ergebnis der Operation wird die Zahl $Z = (-)\ d_{p_1+p_2-1} \ldots d_{p_1}$ in der Zelle 7735 gespeichert. Im Register R steht sie um 8 Dualstellen links verschoben (Speichern in den Adressenstellen). Eventuelle Kapazitätsüberschreitung wird dabei nicht angezeigt.

Das Vorzeichen der ausgeblendeten Zahl Z stimmt mit dem Vorzeichen der Dezimalzahl D überein. Der Inhalt von a* bleibt erhalten.

Bemerkung: Das Eingabebild dieses Externbefehls umfaßt einschließlich WM immer 12 Zeichen.

Beispiel 31: Eine in der Zelle 7000 gespeicherte Dezimalzahl enthält in den Dezimalstellen d_5 bis d_8 eine Information, die in einem bestimmten Zusammenhang die Adresse b darstellt. Es ist der Inhalt der Zelle b in das Indexregister I_5 zu übertragen.

Die Befehlsfolge ist ab 7006 gespeichert.

```
7006 40 17 0 7000. 45  ⟨R⟩ : = b1b2b3b4 000
7007 40  3 0 7741.              (Triadenschreibweise)
7010 40  1 1 0000.       ⟨R⟩ : = b
7011 40  3 0 7745.
```

3.2.23. $S_1S_2 = 40$ unbedingter Sprung

Eingabebild: $4K_2$ 40 I $a_1a_2a_3a_4$ WM

Wirkung: Es wird ein Sprung zur Adresse a^* durchgeführt, d. h., die Programmabarbeitung wird mit dem in der Zelle a^* gespeicherten Befehl fortgesetzt.

Mit Hilfe dieses Befehls kann eine in der natürlichen Reihenfolge ablaufende Befehlsfolge unterbrochen werden.

$\langle PZ \rangle := a^*$

Beispiel 32: In den Zellen 3003 und 3004 stehen die beiden Zahlen D_1 und D_2, in den Zellen 3010 und 3011 die beiden Zahlen D_3 und D_4. Alle Zahlen besitzen zwei Stellen nach dem Komma. Es sind die Ausdrücke
$S_1 = D_1D_2 + D_3$ und $S_2 = S_1 + D_4$ zu bilden und in den Zellen 7000 und 7001 zu speichern. Die Rechnung ist mit 2 Stellen nach dem Komma durchzuführen.

Zur Speicherung der Befehlsfolge stehen die noch freien Zellen zwischen 3000 und 3013 zur Verfügung.

```
3000 40  1 0 3003.
3001 40 15 0 3004. 2   ⟨R⟩ = D1D2 mit 2 Stellen
3002 40 40 0 3005.
3005 40  5 0 3010.
3006 40  3 0 7000.     ⟨7000⟩ = S1
3007 40 40 0 3012.
3012 40  5 0 3011.
3013 40  3 0 7001.     ⟨7001⟩ = S2
```

3.2.24. $S_1S_2 = 31$ Vergleichersprung 1

Eingabebild: $4K_2$ 31 I $a_1a_2a_3a_4$ WM

Wirkung: Bei dem vorliegenden Befehl handelt es sich um einen bedingten Sprungbefehl. Es wird nur dann ein Sprung zur Adresse a^* ausgeführt, wenn der Vergleicher „VG1" gesetzt ist. Andernfalls wird in der natürlichen Reihenfolge weitergearbeitet. Durch die Abfrage des Vergleichszustandes wird der gesetzte Vergleicher nicht zerstört. Es ist eine wiederholte Abfrage möglich.

$\langle PZ \rangle := a^* \,/\, VG1$

Beispiel 33: Die fünf Zahlen $D_1, D_2, \ldots, D_5$ befinden sich in hintereinanderfolgenden Zellen. Die Adresse von D_2 ist in den Adressenstellen von I_4 gespeichert. Die Kommastellung ist bei allen Zahlen dieselbe. Wenn D_1 größer oder gleich D_5 ist, soll der Ausdruck
$S_1 = D_1 + D_2 - D_3 + D_4 - D_5$ gebildet und in der Zelle 6 (relativ) gespeichert werden. Andernfalls ist der Ausdruck $S_2 = D_1 + D_2 - D_3 + D_4 + D_5$ zu bilden und in der Zelle 0 (relativ) zu speichern.

Die Befehlsfolge ist relativ, beginnend bei 1, programmiert

```
  0 0.
→ 1 40  1 4 7777.    ⟨R⟩ := D1
  2 40 10 4 0003.    D5 verglichen mit D1
  3 40  5 4 0000.
  4 40  6 4 0001.    10 41 31 0 0014.
  5 41 40 0 0007.    11 40  5 4 0003.
  6 0.               12 41  3 0 0000.
  7 40  5 4 0002.    14 40  6 4 0003.
                     15 41  3 0 0006.
```

3.2.25. $S_1S_2 = 32$ Vergleichersprung 2

Eingabebild: $4K_2$ 32 I $a_1a_2a_3a_4$ WM

Wirkung: Wenn der Vergleicher „VG2" gesetzt ist, erfolgt ein Sprung zur Adresse a*. Andernfalls wird in der natürlichen Reihenfolge weitergearbeitet.

$\langle PZ \rangle := a^* \;/\; VG2$

Beispiel 34: Die Zahlen von 1 bis 100 sind zu summieren und in I_2 und I_3 zu speichern:

$$S = \sum_{n=1}^{100} n$$

In den Zellen 7000 und 7001 stehen die Konstanten 1 und 100.

Die Befehlsfolge ist relativ programmiert

```
7000  3   1.
7001  3 100.
0000 40  74 1 7742.   n : = 0; s : = 0 (⟨I1⟩ = n / ⟨I2⟩ = S)
   1 40  75 1 7000.   n : = n + 1
   2 40  63 1 7001.   100 verglichen mit n
   3 40  75 2 7741.   S : = S + n
   4 41  32 0 0001.   Sprung bei 100 > n
   5 40  71 3 7742.
```

3.2.26. $S_1S_2 = 33$ Bedingter Sprung – Registerinhalt negativ

Eingabebild: $4K_2$ 33 I $a_1a_2a_3a_4$ WM

Wirkung: Bei diesem bedingten Sprungbefehl erfolgt ein Sprung zur Adresse a*, wenn der Inhalt des Registers R kleiner als Null ist ($z_0 = 1$).

$\langle PZ \rangle := a^* \;/\; \langle R \rangle < 0$

3.2.27. $S_1S_2 = 34$ Bedingter Sprung – Registerinhalt gleich Null

Eingabebild: $4K_2$ 34 I $a_1a_2a_3a_4$ WM

Wirkung: Ist der Inhalt des Registers R gleich Null, erfolgt ein Sprung zur Adresse a*. Andernfalls wird die natürliche Reihenfolge der Befehlsabarbeitung fortgesetzt.

$\langle PZ \rangle := a^* \;/\; \langle R \rangle = 0$

3.2.28. $S_1S_2 = 35$ Bedingter Sprung – Registerinhalt nicht negativ

Eingabebild: $4K_2$ 35 I $a_1a_2a_3a_4$ WM

Wirkung: Es wird ein Sprung zur Adresse a* ausgeführt, wenn der Inhalt des Registers R größer oder gleich Null ist. Andernfalls wird die natürliche Befehlsfolge fortgesetzt.

$\langle PZ \rangle := a^* \;/\; \langle R \rangle \geq 0$

Beispiel 35: In den Zellen 3000 bis 3002 sind die Zahlen D_1, D_2 und D_3 mit 2 Stellen nach dem Komma gespeichert.

Wenn $D_1 = 0$ ist, soll der Ausdruck $S = |D_3|$ gebildet werden; wenn D_1D_2 ($D_1 \neq 0$) größer oder gleich Null ist, soll $S = D_1D_2 - D_3$ und bei D_1D_2 kleiner als Null soll $S = D_1D_2 + D_3$ berechnet und in I_1 gespeichert werden:

$$S = \begin{cases} |D_3| & \text{für } D_1 = 0 \\ D_1D_2 - D_3 & \text{für } D_1D_2 \geq 0 \ (D_1 \neq 0) \\ D_1D_2 + D_3 & \text{für } D_1D_2 < 0 \end{cases}$$

Die Rechnung ist mit 2 Stellen nach dem Komma durchzuführen.

Die Befehlsfolge steht ab 3003.

3003 40 1 0 3000.	$\langle R \rangle := D_1$
3004 40 34 0 3012.	$P \{ D_1 = 0 \}$
3005 40 15 0 3001. 2	
3006 40 35 0 3014.	$P \{ D_1 D_2 \geq 0 \}$
3007 40 5 0 3002.	$\langle R \rangle = D_1 D_2 + D_3$
3010 40 3 0 7741.	$\langle I_1 \rangle = S$
3012 40 12 0 3002.	$\langle R \rangle := \lvert D_3 \rvert$
3013 40 40 0 3010.	
3014 40 6 0 3002.	$\langle R \rangle = D_1 D_2 - D_3$
3015 40 40 0 3010.	

3.2.29. $S_1S_2 = 60$ Bedingter Sprung – Indexregisterinhalt negativ

Eingabebild: $4K_2$ 60 I $a_1a_2a_3a_4$ WM

Wirkung: Bei diesem bedingten Sprungbefehl wird ein Sprung zur Adresse a ausgeführt, wenn der Inhalt des im Befehl angegebenen Indexregisters I_n ($n = 1, \ldots, 7$) negativ ist ($z_0 = 1$). Andernfalls wird in der natürlichen Reihenfolge weitergearbeitet.

$\langle PZ \rangle := a \;/\; \langle I_n \rangle < 0$

3.2.30. $S_1S_2 = 61$ Bedingter Sprung – Indexregisterinhalt gleich Null

Eingabebild: $4K_2$ 61 I $a_1a_2a_3a_4$ WM

Wirkung: Wenn der Inhalt des im Befehl angegebenen Indexregisters I_n ($n = 1, \ldots, 7$) gleich Null ist, erfolgt ein Sprung zur Adresse a. Wenn dies nicht der Fall ist, wird die Befehlsabarbeitung in der natürlichen Reihenfolge fortgesetzt.

$\langle PZ \rangle := a \;/\; \langle I_n \rangle = 0$

3.2.31. $S_1S_2 = 62$ Bedingter Sprung – Indexregisterinhalt nicht negativ

Eingabebild: $4K_2$ 62 I $a_1a_2a_3a_4$ WM

Wirkung: Es wird ein Sprung zur Adresse a ausgeführt, wenn der Inhalt des Indexregisters I_n ($n = 1, \ldots, 7$) größer oder gleich Null ist. Andernfalls wird die natürliche Reihenfolge fortgesetzt.

$\langle PZ \rangle := a \;/\; \langle I_n \rangle \geq 0$

Beispiel 36: Die Zahlen $k_0, k_1, \ldots, k_n$ sind hintereinanderfolgend ab 3000, die Zahlen $I_0, I_1, \ldots, I_n$ hintereinanderfolgend ab 3100 gespeichert. Es ist der Ausdruck

$$S = k_0 I_0 + k_1 I_1 + \ldots + k_n I_n = \sum_{i=0}^{n} k_i I_i$$

zu bilden.

Die Zahlen k_i sind mit 2 Stellen nach dem Komma, die Zahlen I_i mit 3 Stellen gespeichert. Die Summanden und die Summe S sind mit 2 Stellen nach dem Komma zu berechnen. S ist in 2011 zu speichern. Der Parameter n steht in den Adressenstellen von I_1. In den Adressenstellen der Zelle 2010 ist eine Eins gespeichert.

Die Befehlsfolge ist ab 2000 gespeichert.

```
2000 40 74 2 0000.        S : = 0
2001 40  1 1 3000.
2002 40 15 1 3100. 3   ⟨R⟩ : = k_l l_l
2003 40 75 2 7773.        S : = S + k_l l_l
2004 40 76 1 2010.        i : = i — 1
2005 40 62 1 2001.        P (i ≥ 0)
2006 40 73 2 2011.

2010 00  1 000.
2011 0.
```

3.2.32. S_1S_2 = 36 Uberlaufsprung

Eingabebild: $4K_2$ 36 I $a_1a_2a_3a_4$ WM

Wirkung: Bei diesem bedingten Sprungbefehl wird ein Sprung zur Adresse a* ausgeführt, wenn der vorhergehende Befehl ein Additions- oder Subtraktionsbefehl bzw. ein „Negativ-Lesen" ist, der zur Kapazitätsüberschreitung führte. Bei den Externbefehlen gilt dies für die Befehle mit dem Operationscode S_1S_2 = 02/05/06/72/75/76.

Wenn durch diese Befehle kein Überlauf entsteht, wird die Befehlsabarbeitung in der natürlichen Reihenfolge fortgesetzt.

⟨PZ⟩ : = a* / Ü

Beispiel 37: In den Zeilen 3015 bis 3020 sind die Zahlen D_1, D_2, D_3, D_4 gespeichert. Wenn die Summation von D_1 und D_2 zur Kapazitätsüberschreitung führt, soll der Ausdruck $S = D_1 10^{-1} + D_2 10^{-1} + D_3 + D_4$ gebildet werden, wobei die Verschiebung ohne Runden durchzuführen ist. Andernfalls ist $S = D_1 + D_2 + D_3 + D_4$ zu berechnen. Die Summe S ist in 3021 zu speichern.

Die Befehlsfolge ist ab 3000 gespeichert.

```
3000 40  1 0 3015.      ⟨R⟩ : = D1
3001 40  5 0 3016.
3002 40 36 0 3007.      P {Überlauf}
3003 40  5 0 3017.
3004 40  5 0 3020.
3005 40  3 0 3021.      ⟨3021⟩ : = S
3007 40 14 0 3015. 1
3010 40  3 0 3014.      zwischenspeichern
3011 40 14 0 3016.1
3012 40  5 0 3014.
3013 40 40 0 3003.
3014  0.
```

3.2.33. S_1S_2 = 37 Selektorsprung

Eingabebild: $4K_2$ 37 I $a_1a_2a_3a_4$ WM

Wirkung: Es handelt sich um einen bedingten Sprungbefehl, dessen Ausführung von der Stellung des Selektors „SP" abhängig ist. Wenn der Selektor „SP" eingerastet ist, wird ein Sprung zur Adresse a* ausgeführt. Andernfalls wird die Befehlsabarbeitung in der natürlichen Reihenfolge beibehalten.

⟨PZ⟩ : = a* / SP

3.2.34. S_1S_2 = 42 Bedingter Stop

Eingabebild: $4K_2$ 42 | $a_1a_2a_3a_4$ WM

Wirkung: Der Befehl führt zum Stop der Befehlsabarbeitung, wenn der Selektor „BS" gesetzt ist.

Die Programmabarbeitung wird mit einem unbedingten Sprung zur Adresse a* fortgesetzt, wenn der Selektor „BS" gelöst wird oder wenn er nicht gesetzt war bzw. wenn die Starttaste „ST" gedrückt wird.

⟨PZ⟩ : = a* / BS

3.2.35. S_1S_2 = 43 Unbedingter Stop

Eingabebild: $4K_2$ 43 | $a_1a_2a_3a_4$ WM

Wirkung: Der Befehl bewirkt einen Stop der Befehlsabarbeitung. Dieser Zustand kann nur durch Betätigen der Starttaste „ST" oder mit Hilfe der Funktionstasten und -schalter beseitigt werden (s. Heft 3 Bedienungsanleitung). Beim Löschen des Stopzustandes mit Hilfe der Starttaste oder der Tasten- und Schalterkombination „HMR" bzw. „HÜR" wird die Programmabarbeitung mit einem unbedingten Sprung zur Adresse a* fortgesetzt. Bei der Kombination „HGR" gelangt man zur Eingabe eines ersten Kennzeichens K_1.

⟨PZ⟩ : = a* / Stop

3.2.36. S_1S_2 = 50 Anwahl Kanal Null

Eingabebild: $4K_2$ 50 | $a_1a_2a_3a_4$ WM

Wirkung: Es erfolgt eine Anwahl des Kanals Null (Internanwahl). Der Rechner ist dadurch vorbereitet, bei der Aufstellung eines Eingabebefehls Informationen über die Tastatur des Bedienungspultes oder über das Schreibwerk entgegenzunehmen. Die Unterscheidung erfolgt durch den Schalter „T-SW" (s. Heft 3 Bedienungsanleitung). Die Programmabarbeitung wird mit einem Sprung zur Adresse a* fortgesetzt.

⟨PZ⟩ : = a* / AW 0

3.2.37. S_1S_2 = 51 Anwahl Kanal Eins

Eingabebild: $4K_2$ 51 | $a_1a_2a_3a_4$ WM

Wirkung: Die Anwahl des Kanals 1 bewirkt, daß bei der Abarbeitung eines Eingabebefehls die Informationen vom Rechner nur über den Leser 1 entgegengenommen werden können. Bei der Abarbeitung eines Ausgabebefehls erfolgt eine Ausgabe auf Lochband. Die Programmabarbeitung wird mit einem Sprung zur Adresse a* fortgesetzt.

⟨PZ⟩ : = a* / AW 1

3.2.38. S_1S_2 = 52 Anwahl Kanal Zwei

Eingabebild: $4K_2$ 52 | $a_1a_2a_3a_4$ WM

Wirkung: Der Befehl ermöglicht es, bei Aufstellung eines Eingabebefehls die Informationen über den Leser 2 einzulesen. Ein Ausgabebefehl bezieht sich auf das Schreibwerk. Die Programmabarbeitung wird mit einem Sprung zur Adresse a* fortgesetzt.

⟨PZ⟩ : = a* / AW 2

3.2.39. $S_1S_2 = 25$ Eingabe einer Dezimalzahl

Eingabebild: $4K_2$ 25 I $a_1a_2a_3a_4$ WM

Wirkung: Durch diesen Befehl wird vom Rechner eine Dezimalzahl in üblicher Schreibweise eingelesen und in die Zelle a* übertragen. Bei der Eingabe wird das Komma überlesen. Die Zahl wird als ganz interpretiert. Die Kommastellung der einzugebenden Zahl ist vom Programmierer zu berücksichtigen bzw. bei der Erstellung der Daten zu beachten. Die Zahl ist durch WM abzuschließen. Ein positives Vorzeichen ist zu unterdrücken. Bei einer negativen Zahl kann das Vorzeichen an beliebiger Stelle vor dem Schlußzeichen WM gegeben werden.

Somit ist folgende Eingabedarstellung möglich:

$D = (-)\ d_9d_8d_7d_6d_5d_4d_3d_2d_1d_0$ WM

Für die Darstellung einer Null genügt die Angabe der WM. Es können Zahlen aus dem Bereich $|D| \leq 2^{31}-1$ eingegeben werden. Bei Überschreiten dieses Bereiches wird die Zahl verfälscht (vgl. 2.2.6.). Die Dezimalzahl wird bei der Eingabe in die im Abschnitt 1.2. beschriebene Interndarstellung konvertiert.

$\langle a^* \rangle := ED$

Bemerkung: Zur Eingabesteuerung wird das Register R nicht benutzt. Die Anwahl eines Eingabegerätes muß gesondert erfolgen.

Beispiel 40: Die Zahlen $k_0, k_1, \ldots, k_n$ und $l_0, l_1, \ldots, l_n$ sind auf verschiedenen Lochbändern abgelocht. Es ist der Ausdruck

$$S = k_0l_0 + k_1l_1 \ldots + k_n l_n = \sum_{i=0}^{n} k_i l_i$$

zu bilden und in l_1 zu speichern. Bei den Zahlen k_i sind 2 Stellen nach dem Komma berücksichtigt, bei den Zahlen l_i 3 Stellen. Die k_i sind über den Leser 2, die l_i über den Leser 1 einzugeben. Es ist jeweils ein Paar (k_i, l_i) einzugeben. Die Summanden und die Summe S sind mit 2 Stellen nach dem Komma zu berechnen. Das Programm ist relativ programmiert.

```
0 40 74 1 0000.     S : = 0
1 41 52 0 0002.     AW 2
2 40 25 0 7773.     ⟨R⟩ : = k_i
3 41 51 0 0004.     AW 1
4 40 25 0 7742.     ⟨l_2⟩ : = l_i zwischenspeichern
5 40 15 0 7742. 3   ⟨R⟩ = k_i l_i
6 40 75 1 7773.     S : = S + k_i l_i
7 41 40 0 0001.
```

Bemerkung: Nach dem Verarbeiten des letzten Paares bleibt der Rechner im Eingabehalt stehen, da keine Zahl mehr bereitgestellt ist.

3.2.40. $S_1S_2 = 26$ Eingabe von Dezimalzahlen im Satz

Eingabebild: $4K_2$ 26 I $a_1a_2a_3a_4$ WM

Wirkung: Durch diesen Befehl können Dezimalzahlen entsprechend der Darstellung im Abschnitt 3.2.39. vom Rechner eingelesen und in den Hauptspeicher übertragen werden. Die Speicherung der Zahlen erfolgt fortlaufend ab der Zelle a*. Zur Beendigung der Eingabe stehen in der 8-Kanal-Variante drei Marken (Satzmarke „SM/WRZL", Blockmarke „BM", Doppelpunkt „:") und

in der 5-Kanal-Variante eine Marke (Doppelpunkt „:") zur Verfügung. Alle Marken sind gleichberechtigt. Keine ist einer anderen übergeordnet. Diese Schlußzeichen schließen die Funktion von WM nicht ein. Die Fortsetzung des Programmes nach dem Einlesen eines Satzes ist in Abhängigkeit von der Schlußmarke folgendermaßen bestimmt:

Der Eingabebefehl stehe in der Zelle b.

8-Kanal: Bei Abschluß des Satzes durch „SM" wird das Programm mit dem Befehl in der Zelle b + 1 fortgesetzt, bei Abschluß durch „BM" oder „:" mit dem Befehl in der Zelle b + 2.

5-Kanal: Das Programm wird mit dem Befehl in der Zelle b + 1 fortgesetzt.

Ein Satz von Dezimalzahlen kann in der 8-Kanal-Variante durch die Zeichen „Irrung Satz", „Irrung Block", Apostroph „'" und in der 5-Kanal-Variante durch das Zeichen „H" gelöscht werden. Ein Irrungszeichen kann an einer beliebigen Stelle eines Satzes, aber vor dem entsprechenden Satzschlußzeichen, gegeben werden.

$\langle a^* + i \rangle := ES$

Bemerkung: Zur Eingabesteuerung wird das Register R nicht benutzt. Die Anwahl eines Eingabegerätes muß gesondert erfolgen.

Beispiel 41: Auf einem Lochband sind n Sätze zu je drei Zahlen k_i, l_i, m_i $(i = 1, \ldots, n)$ abgelocht.

Es ist der Ausdruck

$$S = \frac{k_1 l_1}{m_1} + \frac{k_2 l_2}{m_2} + \ldots + \frac{k_n l_n}{m_n} = \sum_{i=1}^{n} \frac{k_i l_i}{m_i}$$

zu bilden und in der Zelle 6000 zu speichern. Die Zahlen k_i und m_i sind mit 2 Stellen nach dem Komma abgelocht, die Zahlen l_i mit 3 Stellen. Die Zwischenwerte und die Summe S sind auf 2 Stellen genau zu berechnen. Ein Satz wird durch die Marke „SM" abgeschlossen. Als Schlußzeichen der n-Sätze wird „BM" verwendet. Zur Eingabe wird der Leser 2 benutzt. Das Beispiel bezieht sich auf die 8-Kanal-Variante. Die Befehlsfolge ist ab 7001 gespeichert.

```
7001 40 52 0 7002.     AW 2
7002 40 74 4 0000.     S : = 0
7003 40 26 0 7741:     ES
7004 40 40 0 7007.
7005 40 73 4 6000.     ⟨6000⟩ : = S
7006 40 43 0 0000.
7007 40  1 0 7741.     ⟨R⟩ : = k_i
7010 40 15 0 7742. 3
7011 40 16 0 7743. 2
7012 40 75 4 7773.     S : = S + k_i l_i / m_i
7013 40 40 0 7003.
```

3.2.41. $S_1S_2 = 27$ Eingabe eines Wortes mit Kennzeichen

Eingabebild: $4K_2$ 27 l $a_1a_2a_3a_4$ WM

Wirkung: Der Befehl ermöglicht die Eingabe eines Wortes mit Eingabekennzeichen für $K_1 = 0$ bis $K_1 = 5$.

Die Eingabeorganisation wurde im Abschnitt 2.2. dargestellt.

Worte mit Kennzeichen $K_1 = 0, 3, 4$ werden in die jeweilige Interndarstellung konvertiert und in der Zelle a* gespeichert.

Ein Sofortbefehl ($K_1 = 1$, $K_1 = 5$) wird ausgeführt.

Eine Adressenänderung 1. Art ist nur bei bereitgestellten Leitadressen möglich. Die 2. bis 9. Leitadresse können vor einem Wort mit $K_1 = 0, 1, 3, 4, 5$ gegeben werden. Die Eingabe der Relativadresse und der 1. LA ist verboten, wenn das nachfolgende einzugebende Wort gespeichert werden soll.

Bemerkung: Zur Eingabesteuerung wird das Register R nicht benutzt. Die Anwahl eines Eingabegerätes muß gesondert erfolgen.

Beispiel 42: Die Aufgabenstellung unterscheidet sich von der des Beispieles 40 dadurch, daß die Zahlen k_i in Triadenschreibweise mit den entsprechenden Eingabekennzeichen ($K_1 = 0$, $K_2 = 0$) vorliegen.

Das Programm ist relativ programmiert.

```
0 40 74 1 0000.    S : = 0
1 41 52 0 0002.    AW 2
2 40 27 0 7773.    Eingabe Wort mit Kennzeichen
3 41 51 0 0004.    AW 1
4 40 25 0 7742.    ED
5 40 15 0 7742. 3
6 40 75 1 7773.    S : = S + k_i l_i
7 41 40 0 0001.
```

3.2.42. $S_1S_2 = 30$ Eingabe eines Programmteiles

Eingabebild: $4K_2$ 30 I $a_1a_2a_3a_4$ WM

Wirkung: Durch diesen Befehl wird die Regie der im 2. Kapitel beschriebenen Eingabeorganisation übergeben. Es können Informationen mit Eingabekennzeichen entsprechend der dort angeführten Organisation eingegeben werden, z. B. Leitadressen, Zahlen, Programmteile.

Bei der Eingabe von abzuspeichernden Worten müssen die benötigten Leitadressen bereitgestellt sein. Dies kann durch die Eingabe der Leitadressen mit $K_1 = 2$ vor Eingabe der abzuspeichernden Worte erfolgen.

Die Adresse a* stellt die Adresse a der 1. LA dar. Sie wird bei der Eingabe einer neuen 1. LA überschrieben. Die Programmabarbeitung wird mit dem Befehl in der Zelle b + 1 fortgesetzt, wenn der Befehl „Eingabe eines Programmteiles" in b gespeichert ist und als 1. Kennzeichen K_1 = „BM" (Blockmarke) oder K_1 = „:" bei der 8-Kanal-Version bzw. K_1 = „:" bei der 5-Kanal-Version gegeben wird.

Bemerkung: Zur Eingabesteuerung wird das Register R nicht benutzt. Die Anwahl eines Eingabegerätes muß gesondert erfolgen.

Beispiel 43: Zur Speicherung eines Programmes steht nur eine Spur, z. B. die Zellen 3000 bis 3037 (oktal), zur Verfügung.

Das Programm ist auf einem Lochband zu Blöcken von je 30 Worten (dezimal) abgelocht. Ein Block ist durch das 1. Kennzeichen K_1 = „:" abgeschlossen. Eine Adressenänderung 1. Art soll nur mit Hilfe der Adresse a der 1. LA durchgeführt werden.

Die Eingabe erfolgt über den Leser 1.

In den Zellen 3000 und 3001 sind folgende Befehle gespeichert:

```
3000 40 51 0 3001.    AW 1
3001 40 30 0 3002.    Eingabe eines Programmteiles
```

Nach der Anwahl des Lesers 1 erfolgt die Eingabe eines Blockes durch den Befehl in der Zelle 3001. Nach der Übertragung in die Zellen 3002 bis 3037 wird dieser, beginnend mit dem Befehl in 3002, abgearbeitet. Wenn der letzte auszuführende Befehl eines Blockes ein unbedingter Sprung zur Zelle 3000 ist, wird anschließend der nächste Block eingelesen und danach abgearbeitet usw. Das Programm muß bei diesem Verfahren weitgehend gestreckt programmiert sein. Der Anwahlbefehl in der Zelle 3000 muß bei der Programmierung entsprechend berücksichtigt werden.

3.2.43. S_1S_2 = 53 Wagenrücklauf

Eingabebild: $4K_2$ 53 I $a_1a_2a_3a_4$ WM

Wirkung: Bei Anwahl des Schreibwerkes wird ein Wagenrücklauf ohne Zeilenvorschub durchgeführt.
Bei Anwahl des Stanzers wird sowohl in der 8-Kanal- als auch in der 5-Kanal-Version das entsprechende Zeichen des externen R 300-Codes gestanzt.
Die Befehlsabarbeitung wird mit einem unbedingten Sprung zur Adresse a* fortgesetzt.
⟨PZ⟩ : = a* / WR

Bemerkung: Zur Ausführung eines „WR" muß der Wagen mindestens 3 Stellen vom linken Randsteller entfernt stehen.

3.2.44. S_1S_2 = 54 Wagenrücklauf mit Zeilenvorschub

Eingabebild: $4K_2$ 54 I $a_1a_2a_3a_4$ WM

Wirkung: Bei Anwahl des Schreibwerkes wird ein Wagenrücklauf mit Zeilenvorschub durchgeführt
Bei Anwahl des Stanzers wird sowohl in der 8-Kanal- als auch in der 5-Kanal-Version das entsprechende Zeichen des externen R 300-Codes gestanzt
Die Befehlsabarbeitung wird mit einem unbedingten Sprung zur Adresse a* fortgesetzt.
⟨PZ⟩ : = a* / WRZL

Bemerkung: Zur Ausführung eines „WRZL" muß der Wagen mindestens 3 Stellen vom linken Randsteller entfernt stehen.

3.2.45. S_1S_2 = 55 Zeilenvorschub

Eingabebild: $4K_2$ 55 I $a_1a_2a_3a_4$ WM

Wirkung: Bei Anwahl des Schreibwerkes wird ein Zeilenvorschub durchgeführt. Bei Anwahl des Stanzers wird sowohl in der 8-Kanal- als auch in der 5-Kanal-Version das entsprechende Zeichen des externen R 300-Codes gestanzt. Die Befehlsabarbeitung wird mit einem unbedingten Sprung zur Adresse a* fortgesetzt.
⟨PZ⟩ : = a* / ZL

3.2.46. S_1S_2 = 56 Zwischenraum

Eingabebild: $4K_2$ 56 I $a_1a_2a_3a_4$ WM

Wirkung: Bei Anwahl des Schreibwerkes wird ein Leerschritt ausgeführt. Bei Anwahl des Stanzers wird sowohl in der 8-Kanal- als auch in der 5-Kanal-Version das entsprechende Zeichen des externen R 300-Codes gestanzt. Die Befehlsabarbeitung wird mit einem unbedingten Sprung zur Adresse a* fortgesetzt.
⟨PZ⟩ : = a* / ZR

3.2.47. S_1S_2 = 57 Tabulator

Eingabebild: $4K_2$ 57 I $a_1a_2a_3a_4$ WM

Wirkung: Bei Anwahl des Schreibwerkes wird ein Tabulatorsprung durchgeführt. Bei Anwahl des Stanzers wird sowohl in der 8-Kanal- als auch in der 5-Kanal-Version das entsprechende Zeichen des externen R 300-Codes gestanzt.

Die Programmabarbeitung wird mit einem unbedingten Sprung zur Adresse a* fortgesetzt.

⟨PZ⟩ : = a* / Tab

3.2.48. S_1S_2 = 20 Ausgabe einer Dezimalzahl / Variante 1

Eingabebild: $4K_2$ 20 I $a_1a_2a_3a_4$ WM $p_3p_2p_1$

Wirkung: Der Inhalt der Zelle a* wird als Dezimalzahl D = (—) $d_9d_8d_7d_6d_5d_4d_3d_2d_1d_0$ interpretiert. In der Operationsergänzung ist die Angabe dreier Parameter notwendig. Sie dienen der Druckaufbereitung. Der Parameter p_1 (p_1 = 0, 1, ..., 9) gibt die erste auszugebende Dezimalstelle d_{p_1} an, p_2 (p_2 = 1, ..., 9) bestimmt die Anzahl der Stellen vor und p_3 (p_3 = 0, ..., 7) die Anzahl der Stellen nach dem Komma.

Die auszugebende Zahl wird gerundet. Bei der Ausgabe werden die durch die Angabe der Parameter ausgeschlossenen Dezimalstellen nicht berücksichtigt. Das Vorzeichen der auszugebenden Zahl ist durch das Vorzeichen der Dezimalzahl D bestimmt. Es wird am Ende einer Zahl gegeben. Bei einer positiven Zahl wird ein Leerschritt ausgeführt. Vornullen werden durch Zwischenräume ersetzt. Bei Zahlen, die betragsmäßig kleiner als 1 sind, wird mit „0," begonnen.

Der Inhalt der Zelle a* wird nicht zerstört.

Bemerkung: Die Rundung wird mit der Dezimalzahl D durchgeführt. Dabei kann es zur Kapazitätsüberschreitung und damit zum Überlauf kommen. Zur Ausgabesteuerung wird das Register R nicht benutzt. Eine Umschaltung auf kleine Buchstaben ist nicht berücksichtigt.

Die Anwahl eines Ausgabegerätes muß gesondert erfolgen.

Bei angewähltem Stanzer wird die Zahl sowohl in der 8-Kanal- als auch in der 5-Kanal-Variante im R 300-Code gestanzt.

Das Eingabebild dieses Externbefehls umfaßt einschließlich WM immer 13 Zeichen.

Beispiel 44: In der Zelle 2016 ist die Zahl D_1 = −1234560097;
in der Zeile 2017 die Zahl D_2 = 18099998
gespeichert.

Die Dezimalstellen d_6 bis d_9 und d_0 bis d_3 der Zahl D_1 sind mit 2 Stellen nach dem Komma auszudrucken; die Dezimalstellen d_5 und d_6 von D_1 und die Dezimalstellen d_6 bis d_9 sowie d_2 bis d_5 der Zahl D_2 sind mit einer Stelle nach dem Komma auszudrucken. Außerdem ist die in der Dezimalstelle d_0 von D_2 gespeicherte Ziffer als Zehnerstelle einer Zahl aufzufassen und mit 2 Stellen nach dem Komma auszudrucken. Die Zahlen sind untereinander zu schreiben.

Die Befehlsfolge ist ab 2000 gespeichert.

2000	40	52	0	2001.	AW 2							
2001	40	54	0	2002.	Druck	1	2	,	3	5	–	
2002	40	20	0	2016.229	WRZL							
2003	40	54	0	2004.								
2004	40	20	0	2016.223	Druck	(ZR)	0	,	9	7	–	
2005	40	54	0	2006.								
2006	40	20	0	2016.116	Druck	4	,	6	–			
2007	40	54	0	2010.								
2010	40	20	0	2017.139	Druck	(ZR)	(ZR)	1	,	8	(ZR)	
2011	40	54	0	2012.								
2012	40	20	0	2017.135	Druck	1	0	0	,	0	(ZR)	
2013	40	54	0	2014.								
2014	40	20	0	2017.220	Druck	8	0	,	0	0	(ZR)	
2015	40	43	0	0000.								
2016	3	–1234560097.										
2017	3	18099998.										

3.2.49. $S_1S_2 = 21$ Ausgabe einer Dezimalzahl / Variante 2

Eingabebild: $4K_2$ 21 I $a_1a_2a_3a_4$ WM $p_3p_2p_1$

Wirkung: Bis auf die Unterdrückung der Rundung hat dieser Befehl dieselbe Wirkung wie der im vorhergehenden Abschnitt 3.2.48. beschriebene Befehl „Ausgabe einer Dezimalzahl / Variante 1".

3.2.50. $S_1S_2 = 22$ Ausgabe einer Dezimalzahl / Variante 3

Eingabebild: $4K_2$ 22 I $a_1a_2a_3a_4$ WM $p_3p_2p_1$

Wirkung: Gegenüber dem im Abschnitt 3.2.48. beschriebenen Befehl „Ausgabe einer Dezimalzahl / Variante 1" ergeben sich folgende Änderungen:

a) Bei einer positiven Zahl wird der Leerschritt am Ende der Zahl unterdrückt.

b) Eine Zahl wird durch „WM" abgeschlossen. In der 8-Kanal-Variante wird dabei das Zeichen „Tab", in der 5-Kanal-Variante das Zeichen Punkt „." verwendet.

Bemerkung: Bei Ausgabe auf Lochband wird die Zahl in einer Form gestanzt, in der sie ohne Eingabekennzeichen wieder vom Rechner eingelesen werden kann.

In der 5-Kanal-Variante wird dieser Befehl durch ein Zusatzprogramm, das auf beliebig freie Speicherplätze gebracht werden kann, realisiert. In diesem Fall ist nur die Ausgabe auf Lochband sinnvoll.

Beispiel 45: Die im Beispiel 41 angegebene Aufgabe wird erweitert. Die Zahlen k_i, l_i, m_i und der Ausdruck $S_i = \frac{k_i l_i}{m_i}$ eines Satzes sollen jeweils in einer Zeile ausgedruckt werden. Dabei sind sie durch einen Tabulatorsprung zu trennen. Nach dem Druck von S_i ist ein WRZL durchzuführen. Die Summe S ist in der Spalte S_i zu drucken. Außerdem ist sie mit WM auszustanzen. Die Zahlen k_i

mögen max. 3 Stellen vor dem Komma besitzen, die Zahlen l_i und m_i max. 2 Stellen. Außerdem seien die Zahlen $m_i \geq 1$. Die Summe S überschreite nicht 6 Stellen vor dem Komma.

Das aufgestellte Programm bezieht sich auf die 8-Kanal-Variante. Es ist relativ programmiert.

```
 0 41 52 0 0001.      AW 2
 1 41 54 0 0002.      WRZL
 2 40 74 4 0000.      S : = 0
 3 40 26 0 7741.      ES
 4 41 40 0 0015.
 5 41 57 0 0006.      Tab
 6 40 73 4 6000.      ⟨6000⟩ : = S.
 7 41 57 0 0010.      Tab
10 41 57 0 0011.      Tab
11 40 21 0 7744.267   Druck S
12 41 51 0 0013.      AW 1
13 40 22 0 7744.267   Stanzen S
14 40 43 0 0000.
15 40  1 0 7741.      ⟨R⟩ : = k_i
16 40 22 0 7773.234   Druck k_i und Tab
17 40 15 0 7742.3
20 40 22 0 7742.324   Druck l_i und Tab
21 40 16 0 7743.2
22 40 22 0 7743.223   Druck m_i und Tab
23 40 75 4 7773.      S : = S + S_i
24 40 21 0 7773.267   Druck S_i
25 41 54 0 0003.      WRZL
```

3.2.51. $S_1S_2 = 23$ Ausgabe von Triaden mit Eingabekennzeichen

Eingabebild: $4K_2$ 23 I $a_1a_2a_3a_4$ WM

Wirkung: Durch diesen Befehl wird der Inhalt der Zelle a^* in Form von 11 aufeinanderfolgenden Triaden ausgegeben. Je drei duale Wortstellen sind dabei zu einer Oktalziffer zusammengefaßt. Es werden die Eingabekennzeichen $K_1 = 0$ und $K_2 = 0$ mit ausgegeben. Ein Wort wird durch „WM" abgeschlossen. In der 8-Kanal-Variante wird dabei das Zeichen „Tab" und in der 5-Kanal-Variante das Zeichen Punkt „." verwendet.

Bemerkung: Zur Ausgabesteuerung wird das Register R nicht benutzt. Die Anwahl eines Ausgabegerätes muß gesondert erfolgen. Bei Ausgabe auf Lochband wird das Wort in einer Form gestanzt, in der es wieder vom Rechner eingelesen werden kann. In der 5-Kanal-Variante wird dieser Befehl durch ein Zusatzprogramm realisiert. Dieses Programm kann auf beliebig freie Speicherplätze gebracht werden. In diesem Fall ist nur die Ausgabe auf Lochband sinnvoll.

Beispiel 46: Die im Beispiel 41 angegebene Aufgabe wird folgendermaßen erweitert:

Die Teilsummen $S_i = \frac{k_i l_i}{m_i}$ sind in Triadenschreibweise auszustanzen.

Die Summe S ist mit 6 Stellen vor und 2 Stellen nach dem Komma auszudrucken. Zur Eingabe wird der Leser 1 benutzt. Das Programm ist relativ ab 1 programmiert.

```
 1 41 51 0 0002.     AW 1
 2 40 74 4 0000.     S : = 0
 3 42 26 0 0000.     ES
 4 41 40 0 0012.
 5 41 52 0 0006.     AW 2
 6 40 73 0 6000.     ⟨6000⟩ : = S
 7 41 54 0 0010.     WRZL
10 40 21 0 7744.267  Druck S
11 40 43 0 0000.
12 42  1 0 0000.     ⟨R⟩ : = k_i
13 42 15 0 0001.3
14 42 16 0 0002.2
15 40 23 0 7773.     Stanzen S_i
16 40 75 4 7773.     S : = S + S_i
17 41 40 0 0003.
```

Bemerkung: Die Speicherzellen für die Zahlen eines Satzes werden durch die 2. LA bestimmt.

3.2.52. $S_1S_2 = 24$ Textausgabe

Eingabebild: $4K_2$ 24 I $a_1a_2a_3a_4$ WM

Wirkung: Der Befehl ermöglicht die Ausgabe von alphanumerischen Zeichen, die entsprechend der im Abschnitt 1.4. angegebenen Darstellung aufbereitet sind.

Die Adresse a^* gibt das erste zu entschlüsselnde Textwort an. Nach dessen Verarbeitung wird der Inhalt des darauffolgenden Speicherplatzes als Textwort interpretiert usw.

Als Ausgabeende wird die Kombination „77" gewertet.

Bemerkung: Zur Ausgabesteuerung wird das Register R nicht benutzt. Die Anwahl eines Ausgabegerätes muß gesondert erfolgen. Beim Stanzen erfolgt die Codierung der Zeichen im R 300-Code.

Beispiel 47: Auf einem Lochband sind n Sätze zu je drei Zahlen k_i, l_i, m_i $(i = 1, \ldots, n)$ abgelocht. Dabei stellt k_i eine Artikelnummer – ganze Zahl, max. 9 Stellen –, l_i eine Menge in kg – 3 Stellen nach dem Komma, max. 3 Stellen vor dem Komma – und m_i den zugehörigen Einzelpreis in M – 2 Stellen nach dem Komma, max. 3 Stellen vor dem Komma – dar. Ein Satz ist durch die Satzmarke „SM" abgeschlossen.

Als Schlußzeichen der n Sätze wird das Zeichen Blockmarke „BM" verwendet. (Das Beispiel bezieht sich auf die 8-Kanal-Version.) Die Artikelnummer, die Menge, der Einzelpreis und der Gesamtpreis $S_i = l_i\, m_i$ eines Satzes sind jeweils in einer Zeile auszudrucken. Außerdem ist die Summe $S = \sum_{i=1}^{n} l_i\, m_i$ in der letzten Spalte anzugeben.

Der Gesamtpreis S_i und die Summe S sind mit 2 Stellen nach dem Komma zu berechnen und mit 7 Stellen vor und 2 Stellen nach dem Komma zu drucken. Im Kopf der Tabelle sind die Bezeichnungen „Artikelnummer", „Menge/kg", „Einzelpreis/M" und „Gesamtpreis/M" anzugeben. Die Summe S ist durch den Druck von „Summe" zu kennzeichnen.

Die Befehlsfolge steht ab 2000. Die alphanumerischen Zeichen sind ab 2020 bzw. 2035 gespeichert. Zur Eingabe wird der Leser 2 benutzt.

2000 40 52 0 2001.	AW 2
2001 40 24 0 2020.	Druck des Kopfes
2002 40 74 4 0000.	S : = 0
2003 40 26 0 7741.	ES
2004 40 40 0 2010.	
2005 40 24 0 2035.	Druck von Schlußstrich und „Summe"
2006 40 21 0 7744.278	Druck S
2007 40 43 0 0000.	
2010 40 1 0 7742.	⟨R⟩ : = l_1
2011 40 22 0 7741.098	Druck k_1 und Tab
2012 40 15 0 7743.3	⟨R⟩ = $l_1 m_1$
2013 40 22 0 7742.335	Druck l_1 und Tab
2014 40 75 4 7773.	S : = S + S_1
2015 40 22 0 7743.234	Druck m_1 und Tab
2016 40 21 0 7773.278	Druck Si
2017 40 54 0 2003.	

2020 00 32 37 21 57 51.	WRZL	GB	A	KB	r
2021 00 63 31 42 25 43.	t	i	k	e	l
2022 00 45 64 44 44 25.	n	u	m	m	e
2023 00 51 75 37 44 57.	r	Tab	GB	M	KB
2024 00 25 45 27 25 61.	e	n	g	e	/
2025 00 42 27 75 37 25.	k	g	Tab	GB	E
2026 00 57 31 45 71 25.	KB	i	n	z	e
2027 00 43 47 51 25 31.	l	p	r	e	i
2030 00 62 61 37 44 75.	s	/	GB	M	Tab
2031 00 27 57 25 62 21.	G	KB	e	s	a
2032 00 44 63 47 51 25.	m	t	p	r	e
2033 00 31 62 61 37 44.	i	s	/	GB	M
2034 00 57 32 72 77 00.	KB	WRZL	ZL	77	
2035 00 75 75 75 37 20.	Tab	Tab	Tab	GB	=
2036 00 20 20 20 20 20.	=	=	=	=	=
2037 00 20 20 20 20 32.	=	=	=	=	WRZL
2040 00 75 75 62 57 64.	Tab	Tab	S	KB	u
2041 00 44 44 25 15 75.	m	m	e	:	Tab
2042 00 77 00 00 00 00.	77				

3.2.53. S_1S_2 = 41 Unterprogrammsprung

Eingabebild: $4K_2$ 41 I $a_1a_2a_3a_4$ WM

Wirkung: Im Indexregister I_7 wird ein Externbefehl „unbedingter Sprung zur Adresse b + 1" (40 0 b + 1) aufgebaut. b + 1 stellt dabei den um eine Einheit erhöhten Inhalt des Pseudozählers dar. Der Befehl „Unterprogrammsprung" steht dabei in der Zelle b.

Nach Aufbau des Befehls in I_7 erfolgt ein unbedingter Sprung zur Adresse a*.

⟨I_7⟩ : = ⟨PZ⟩ + 1

⟨PZ⟩ : = a*

Beispiel 48: Mit Hilfe dieses Befehls kann eine Unterprogrammtechnik realisiert werden. Der in der Zelle a* gespeicherte Befehl stellt den ersten abzuarbeitenden Befehl des Unterprogramms dar. Eine Rückkehr in das Hauptprogramm ist möglich, wenn als letzter Befehl des Unterprogramms der aufgebaute Befehl „unbedingter Sprung zur Adresse b + 1" ausgeführt wird.

Wird der Inhalt von I_7 durch das Unterprogramm nicht zerstört, kann dieser als letzter abzuarbeitender Befehl des Unterprogramms interpretiert werden. Der vorletzte Befehl muß dann ein unbedingter Sprung zur Zelle 7747 sein.

Wird im Unterprogramm der Inhalt von I_7 zerstört, z. B. bei verschachtelten Unterprogrammen, muß der aufgebaute Sprungbefehl vorher gerettet werden. Dies kann durch Umspeichern des Indexregisterinhaltes in eine Zelle a erfolgen (Externbefehl $S_1S_2 = 73$). Der Anschluß an das Hauptprogramm ergibt sich, indem als letzter Befehl des Unterprogramms der Inhalt von a interpretiert wird.

4. Programmierung in der Maschinensprache

Die Externbefehle, die nicht direkt auf verdrahteten Operationen aufbauen, werden beim FIPS I durch Unterprogramme realisiert. Diese Programme können auch durch einen Maschinenbefehl „Sprung mit Rückkehrabsicht" aufgerufen werden. Damit ist es möglich, im System FIPS I Programme in der Maschinensprache aufzustellen und auf die verdrahtete Arbeitsweise zurückzugreifen. Eine solche Programmierung ist dann notwendig, wenn die interpretierende Abarbeitung von Befehlen zeitlich ungünstig ist.

Da die Versorgungszellen der Unterprogramme des FIPS I bei der Programmierung in der Maschinensprache durch das entsprechende Objektprogramm gefüllt werden müssen, kann sich ein relativ hoher Speicherplatzbedarf für das Objektprogramm ergeben.

In diesem Kapitel werden die Anschlußbedingungen für die Unterprogramme des FIPS I bei der Programmierung in der Maschinensprache angegeben. Die Einsprungadressen sind im allgemeinen in der 8-Kanal- und 5-Kanal-Variante verschieden. Ist beim entsprechenden Eingabebild nur ein Befehl angegeben, stimmen die Einsprungadressen überein.

Die Wirkung der Unterprogramme ist weitgehend bei der Beschreibung der entsprechenden Externbefehle angegeben.

4.1. Eingabeprogramme

Beim C 8205 ist nur die Eingabe eines einzigen Zeichens verdrahtet. Für die Eingabe von Worten unterschiedlichster Bedeutung sind spezielle Leseprogramme erforderlich. Diese können durch einen entsprechenden Unterprogrammaufruf ausgewählt werden.

Die Anwahl eines Eingabegerätes ist in den Leseprogrammen nicht enthalten.

4.1.1. Eingabe einer Dezimalzahl

Eingabebild: 00 254 025 WM

Versorgungszelle: keine

Wirkung: Vom Rechner wird eine Dezimalzahl in üblicher Schreibweise eingelesen. Nach dem Rücksprung steht diese, in die Interndarstellung konvertiert, im Maschinenakkumulator AC.

Die Eingabedarstellung und der Zahlenbereich sind im Abschnitt 3.2.39. (Externbefehl $S_1S_2 = 25$) angegeben.

$\langle AC \rangle := ED$

4.1.2. Eingabe von Dezimalzahlen im Satz

Eingabebild: 00 163 025 WM 8-Kanal
00 165 025 WM 5-Kanal

Versorgungszelle: 7706 00 $a_1a_2a_3a_4$ 240

Wirkung: Durch diesen Befehl können Dezimalzahlen in üblicher Schreibweise vom Rechner eingelesen und in den Hauptspeicher übertragen werden. Die Speicherung der Zahlen erfolgt fortlaufend ab der Zelle $a = a_1a_2a_3a_4$. Zu diesem Zweck muß der Speicherplatz 7706 vorher mit einem entsprechenden Abspeicherbefehl gefüllt worden sein. Die Eingabedarstellung und die Fortsetzung des Hauptprogramms bei Eingabeende ist im Abschnitt 3.2.40. (Externbefehl $S_1S_2 = 26$) angegeben.

$\langle a + i \rangle := ES$

Bemerkung: Der Abspeicherbefehl wird nach der Eingabe eines Wortes jeweils um eine Einheit in der Adresse erhöht.

4.1.3. Eingabe eines Wortes mit Kennzeichen

Eingabebild: 00 213 025 WM 8-Kanal
00 222 025 WM 5-Kanal

Versorgungszellen: keine

Wirkung: Der Befehl ermöglicht die Eingabe eines Wortes mit Eingabekennzeichen für $K_1 = 0$ bis $K_1 = 5$. Die Eingabeorganisation wurde im Abschnitt 2.2. dargestellt. Worte mit Kennzeichen $K_1 = 0, 3, 4$ stehen nach dem Rücksprung, in die jeweilige Interndarstellung konvertiert, im Maschinenakkumulator AC.

Eine mögliche Adressenänderung 1. Art ist im Abschnitt 3.2.41. (Externbefehl $S_1S_2 = 27$) beschrieben.

4.1.4. Eingabe eines Programmteiles

Eingabebild: 00 234 025 WM 8-Kanal
00 142 025 WM 5-Kanal

Versorgungszellen:

7721	1. LA
7722	2. LA
.	.
.	.
.	.
7731	9. LA

Wirkung: Durch diesen Befehl wird die Regie dem im Abschnitt 2.2. beschriebenen „Eingabeprogramm" übergeben. Es können Informationen mit Eingabekennzeichen entsprechend der dort angeführten Organisation eingegeben werden, z. B. Zahlen, Programmteile, Leitadressen. Bei der Eingabe von abzuspeichernden Worten müssen die benötigten Leitadressen bereitgestellt sein. Dies kann durch das Hauptprogramm durch Füllen der Versorgungszellen erfolgen oder durch Eingabe der Leitadressen mit $K_1 = 2$ vor Eingabe der abzuspeichernden Worte.

Die Programmfortsetzung ist im Abschnitt 3.2.42. (Externbefehl $S_1S_2 = 30$) beschrieben.

4.2. Rechenprogramme

Beim C 8205 sind nur die Festkommaaddition und -subtraktion sowie Multiplikationen mit einer Zweierpotenz (binäre Links- und Rechtsverschiebungen) und eine bitweise Konjunktion verdrahtet. Die Multiplikation und Divison müssen durch Unterprogramme realisiert werden. Dies trifft bei der im System „FIPS I" verwendeten Zahlendarstellung auch für Verschiebungen um eine Zehnerpotenz bzw. für die Konjunktion mit einer Dezimalzahl zu.

Vor Aufruf solcher Unterprogramme sind die Operanden und die zugehörigen Parameterangaben vom Hauptprogramm in die Versorgungszellen abzuspeichern. Der Aufruf eines Unterprogrammes braucht nicht direkt nach dem Füllen der Versorgungszellen zu erfolgen, wenn man beachtet, daß deren Inhalte inzwischen nicht zerstört werden.

Bemerkung: Durch die Abarbeitung der Rechenunterprogramme werden die Inhalte der zugehörigen Versorgungszellen i. a. nicht zerstört.

Eine Ausnahme bildet die Konjunktion.

4.2.1. Verschiebung links

Eingabebild: 00 764 025 WM

Versorgungszellen:

7774 D Dezimalzahl

7753 v Verschiebung

Wirkung: Die in der Zelle 7774 stehende Dezimalzahl D wird um v Dezimalstellen (v = 1, ..., 9) links verschoben in den Maschinenakkumulator AC übertragen (Multiplikation mit 10^v).

Der Parameter v ist in den Dualstellen z_{28} bis z_{31} der Zelle 7753 anzugeben (z. B. Eingabe als ganze Zahl). Die anderen Dualstellen können beliebig besetzt sein. Bei Kapazitätsüberschreitung wird das Wort verfälscht. Eine Fehleranzeige ist nicht berücksichtigt.

$\langle AC \rangle := \langle 7774 \rangle \times 10^v$

4.2.2. Verschiebung rechts

Eingabebild: 00 1024 025 WM

Versorgungszellen:

7774 D Dezimalzahl

7753 v Verschiebung

Wirkung: Die in der Zelle 7774 stehende Dezimalzahl D wird um v Dezimalstellen (v = 1, ..., 9) rechts verschoben in den Maschinenakkumulator übertragen (Multiplikation mit 10^{-v}).

Die Verschiebung erfolgt ohne Rundung. Der Parameter v ist in den Dualstellen z_{28} bis z_{31} der Zelle 7753 anzugeben. Die anderen Dualstellen können beliebig besetzt sein.

$\langle AC \rangle := \langle 7774 \rangle \times 10^{-v}$

4.2.3. Multiplikation

Eingabebild: 00 654 025 WM

Versorgungszellen:

7773 Multiplikand

7774 Multiplikator

7753 Verschiebung v

Wirkung: Die in der Zelle 7773 stehende Dezimalzahl wird mit der in der Zelle 7774 gespeicherten Dezimalzahl vorzeichenrichtig multipliziert.

Anschließend an diese Multiplikation kann eine Rechtsverschiebung um v Dezimalstellen (v = 0, ..., 9) durchgeführt werden. Der Parameter v ist in den Dualstellen z_{28} bis z_{31} der Zelle 7753 anzugeben. Die anderen Dualstellen können beliebig besetzt sein. Die Verschiebung wird mit Runden durchgeführt.

Bei Kapazitätsüberschreitung in der einfachen Wortlänge bezüglich des Produktes $\langle 7773 \rangle \times \langle 7774 \rangle \times 10^{-v}$ geht der Automat in Überlauf.

Das Produkt steht nach Beendigung der Operation in AC.

$\langle AC \rangle := \langle 7773 \rangle \times \langle 7774 \rangle \times 10^{-v}$

Bemerkung: Der Verschiebeparameter muß auch bei v = 0 angegeben werden. Das Programm benutzt u. a. die Plätze 7600 bis 7637 als Arbeitszellen.

4.2.4. Division

Eingabebild: 00 637 025 WM

Versorgungszellen:

7773 Dividend
7774 Divisor
7753 Verschiebung v

Wirkung: Die in der Zelle 7773 stehende Dezimalzahl wird durch die in der Zelle 7774 gespeicherte Dezimalzahl vorzeichenrichtig dividiert. Vor Ausführung der Dvision ist es möglich, den Dividenden um v Dezimalstellen ($v = 0, \ldots, 9$) nach links zu verschieben.

Der Verschiebeparameter v ist in den Dualstellen z_{28} bis z_{31} der Zelle 7753 anzugeben. Die anderen Dualstellen können beliebig besetzt sein.

Bei Kapazitätsüberschreitung in der einfachen Wortlänge bezüglich des Quotienten $\langle 7773 \rangle \times 10^v / \langle 7774 \rangle$ geht der Automat in Überlauf.

Der Quotient steht nach der Operation in ⟨AC⟩. Die letzte Dezimalstelle ist gerundet.

$\langle AC \rangle := \langle 7773 \rangle \times 10^v / \langle 7774 \rangle$

Bemerkung: Der Verschiebeparameter muß auch bei $v = 0$ angegeben werden. Das Programm benutzt u. a. die Plätze 7600 bis 7637 als Arbeitszellen.

4.2.5. Konjunktion dezimal

Eingabebild: 00 1236 025 WM

Versorgungszellen:

7774 D Dezimalzahl
7753 p_1 kleinste auszublendende Dezimalstelle
7612 p_2 Anzahl der auszublendenden Ziffern

Wirkung: Durch diesen Befehl kann eine zusammenhängende Folge von Ziffern der in der Zelle 7774 gespeicherten Dezimalzahl $D = (-)\, d_9 d_8 \ldots d_1 d_0$ ausgeblendet werden. Der Parameter p_1 ($p_1 = 0, \ldots, 9$) bestimmt die kleinste Stelle der auszublendenden Zahl. Er ist in den Dualstellen z_{28} bis z_{31} der Zelle 7753 anzugeben. Der Parameter p_2 ($p_2 = 1, \ldots, 9$) gibt die Anzahl der auszublendenden Ziffern an. Er ist in den Dualstellen z_{28} bis z_{31} der Zelle 7612 anzugeben. Die anderen Dualstellen der Zellen 7753 und 7612 können beliebig besetzt sein.

Im Ergebnis der Operation wird die Zahl
$Z = (-)\, d_{p_1 + p_2 - 1} \ldots d_{p_1}$ in der Zelle 7735 gespeichert. In AC steht sie um 8 Dualstellen links verschoben. Eventuelle Kapazitätsüberschreitung wird dabei nicht angezeigt. Das Vorzeichen der ausgeblendeten Zahl stimmt mit dem Vorzeichen der Dezimalzahl D überein.

Bemerkung: Nach Abarbeitung des Unterprogrammes steht in der Zelle 7753 der Parameter p_2.

4.3. Ausgabeprogramme

Beim C 8205 ist nur die Ausgabe eines einzigen Zeichens verdrahtet. Für die Ausgabe von Worten unterschiedlichster Bedeutung sind spezielle Ausgabeprogramme notwendig. Diese können durch einen entsprechenden Unterprogrammaufruf ausgewählt werden.

Vor Aufruf solcher Unterprogramme sind das auszugebende Wort und die zugehörigen Parameterangaben vom Hauptprogramm in die Versorgungszellen abzuspeichern. Der Aufruf eines Unterprogrammes braucht nicht direkt nach dem Füllen der Versorgungszellen zu erfolgen, wenn man beachtet, daß deren Inhalte inzwischen nicht zerstört werden. Die Anwahl eines Ausgabegerätes ist in den Programmen nicht enthalten.

Bemerkung: Durch die Abarbeitung der Unterprogramme werden die Inhalte der zugehörigen Versorgungszellen nicht zerstört.

4.3.1. Ausgabe einer Dezimalzahl / Variante 1

Eingabebild: 00 1144 025 WM

Versorgungszellen:

7774 D Dezimalzahl
7753 p_1 erste auszugebende Dezimalstelle
7612 p_2 Anzahl der Stellen vor dem Komma
7636 p_3 Anzahl der Stellen nach dem Komma

Wirkung: Der Inhalt der Zelle 7774 wird als Dezimalzahl interpretiert. Zur Druckaufbereitung ist die Angabe von drei Parametern notwendig. p_1 ($p_1 = 0, \ldots, 9$) belegt die Dualstellen z_{28} bis z_{31} der Zelle 7753; p_2 ($p_2 = 1, \ldots, 10$) belegt die Dualstellen z_{28} bis z_{31} der Zelle 7612. Der Parameter p_3 ($p_3 = 0, \ldots, 7$) ist in den Stellen z_{29} bis z_{31} der Zelle 7636 zu speichern. Die Belegung der übrigen Dualstellen ist ohne Bedeutung.

Das Ausgabebild ist im Abschnitt 3.2.48. (Externbefehl $S_1S_2 = 20$) angegeben.

Bemerkung: Die Inhalte der Zellen 7612 und 7636 werden durch die Abarbeitung der Unterprogramme Multiplikation und Division zerstört.

4.3.2. Ausgabe einer Dezimalzahl / Variante 2

Eingabebild: 00 1051 025 WM

Versorgungszellen:

7774 D Dezimalzahl
7753 p_1 erste auszugebende Dezimalstelle
7612 p_2 Anzahl der Stellen vor dem Komma
7636 p_3 Anzahl der Stellen nach dem Komma

Wirkung: Bis auf die Unterdrückung der Rundung hat dieser Befehl dieselbe Wirkung wie der im vorhergehenden Abschnitt 4.3.1. beschriebene Befehl.

4.3.3. Ausgabe einer Dezimalzahl / Variante 3

Eingabebild: 00 1211 025 8-Kanal
00 7641 025 5-Kanal

Versorgungszellen:

7774 D Dezimalzahl
7753 p_1 erste auszugebende Dezimalstelle
7612 p_2 Anzahl der Stellen vor dem Komma
7636 p_3 Anzahl der Stellen nach dem Komma

Wirkung: Die Unterschiede in der Wirkung dieses Befehls gegenüber der Wirkung des Befehls „Ausgabe einer Dezimalzahl / Variante 1" sind im Abschnitt 3.2.50. (Externbefehl $S_1S_2 = 22$) angegeben.

4.3.4. Ausgabe von Triaden mit Eingabekennzeichen

Eingabebild: 00 335 025 8-Kanal
00 7642 025 5-Kanal

Versorgungszelle: 7774 auszugebendes Wort

Wirkung: Der Inhalt der Zelle 7774 wird in Form von 11 aufeinanderfolgenden Triaden, versehen mit den Eingabekennzeichen $K_1 = 0$ und $K_2 = 0$ und dem Schlußzeichen „WM", ausgegeben.

Weitere Erläuterungen sind im Abschnitt 3.2.51. (Externbefehl $S_1S_2 = 23$) angegeben.

4.3.5. Textausgabe

Eingabebild: 00 1260 025 WM

Versorgungszelle: 7672 00 $a_1a_2a_3a_4$ 000

Wirkung: Die Wirkung dieses Unterprogramms ist im Abschnitt 3.2.52. (Externbefehl $S_1S_2 = 24$) dargestellt. Dabei gibt die Adresse $a = a_1a_2a_3a_4$ das erste zu entschlüsselnde Textwort an. Zu diesem Zweck muß die Zelle 7672 vom Hauptprogramm gefüllt worden sein.

Beispiel 49: Die im Beispiel 45 gegebene Aufgabe ist in der Maschinensprache zu programmieren.

```
 0 00 4500 030.   AW 2
 1 00 6000 241.   S : = 0
 2 00 2720 030.   WRZL
 3 01   64 211.   ⟨AC⟩ : = 00 7741 240
 4 00 7706 240.
 5 00  163 025.   ES
 6 01   20 000.
 7 00 1550 030.   Tab
10 00 6000 211.
11 00 1550 030.
12 00 7774 240.   ⟨7774⟩ : = S
13 00 1550 030.
14 00 1051 025.   Druck S
15 00 4600 030.   AW 1
16 00 1211 025.   Stanzen S
17 00      020.   Stop
20 00 7741 211.
21 00 7773 240.   ⟨7773⟩ : = k_i
22 00 7774 240.
23 01   60 211.   ⟨AC⟩ : = 00 2150.
24 00 7753 240.   p_1 : = 4
25 00 7626 642.   p_2 : = 3 / p_3 : = 2
26 00 1211 025.   Druck k_i
27 00 7742 211.
30 00 7774 240.   ⟨7774⟩ : = l_i
31 01   63 211.   ⟨AC⟩ : = 00 144.
32 00 7626 642.   p_2 : = 2 / p_3 : = 3
33 00 1211 025.   Druck l_i
34 01   63 211.   ⟨AC⟩ : = 00 2106.
35 00 7753 240.   v = p_1 : = 3
36 00  654 025.
37 00 7773 240.   ⟨R⟩ = k_i l_i
40 01   61 211.   ⟨AC⟩ : = 00 104.
41 00 7612 642.   p_2 : = 2 / p_3 : = 2
42 00 7743 211.
43 00 7774 240.   ⟨7774⟩ : = m_i
44 00 1211 025.   Druck m_i
45 01   41 215.
46 00 7753 240.   v : = 2
47 00  637 025.   ⟨R⟩ = S_i
50 00 7774 240.
51 00 6000 210.
52 00 6000 240.   S : = S + S_i
53 01   62 211.   ⟨AC⟩ : = 00 2316.
54 00 7753 240.   p_1 : = 7
55 00 7626 642.   p_2 : = 6 / p_3 : = 2
56 00 1051 025.   Druck S_i
57 01    2 000.
60 00 2150.
61 00  104.
62 00 2316.
63 00 3106.
64 00 7741 240.
```

5. Zusammenhang zwischen Extern- und Maschinenbefehlen

Durch das im „FIPS I" enthaltene Entschlüsselungsprogramm für Externbefehle können auch Maschinenbefehle interpretiert werden. Voraussetzung hierfür ist, daß in dem zugehörigen gespeicherten Maschinenwort die Oktalstellen t_9 und t_{10} Null sind. Befehle dieser Art können ohne Übergang im Wechsel mit Externbefehlen programmiert werden. Die Interpretation der Maschinenbefehle erfolgt schneller als die Entschlüsselung von Externbefehlen. Im Entschlüsselungsprogramm ergibt sich folgender Befehlsablauf:

```
        .
        .
        .
        00 7773 211.        ⟨AC⟩ : = ⟨R⟩
        00 7753 000.

7753 Maschinenbefehl
7754 00 1357 070.
7755 00 1401 000.

1401 00 7773 240.           ⟨R⟩ : = ⟨AC⟩
1402 00 7751 211.           ⟨AC⟩ : = ⟨PZ⟩
        .
        .
        .
```

Bei interpretierend abzuarbeitenden Maschinenbefehlen tritt an die Stelle des Maschinenakkus AC das Register R. So wird bei Lese- und Transportbefehlen der Inhalt des Registers R mit dem Inhalt der durch die Adresse des Maschinenbefehls bestimmten Zelle verknüpft.

Das Stellen des Maschinenvergleichers durch interpretierend abzuarbeitende Maschinenbefehle ist nicht sinnvoll, da durch die Programme des Systems die Vergleicherstellung im allgemeinen verändert wird.

Verboten ist die Programmierung von X-Befehlen.

Der direkte Anschluß von Unterprogrammen, die in Maschinensprache programmiert sind, ist durch einen Befehl „Sprung mit Rückkehrabsicht" möglich. Dabei wird nach dem Rücksprung der Inhalt von AC ins Register R transportiert. Die Zelle 7753 kann bei Einsprung ins UP nicht zur Parameterübermittlung verwendet werden.

Beispiel 50: In dem im Beispiel 45 angegebenen Programm werden Externbefehle durch Maschinenbefehle, die im Programmablauf dieselbe Wirkung haben, ersetzt.

```
 0 00 4500  030.          AW 2
 1 00 2720  030.          WRZL
 2 00 7744  241.          S : = 0
 3 40 26 0 7741.          ES
 4 41 40 0 0015.
 5 00 1550  030.          Tab
 6 40 73 4 6000.          ⟨6000⟩ : = S
 7 00 1550  030.
10 00 1550  030.
11 40 21 0 7744. 267      Druck S
12 00 4600  030.          AW 1
13 40 22 0 7744. 267      Stanzen S
14 00        020.
15 00 7741  211.          ⟨R⟩ : = k_i
16 40 22 0 7773. 234      Druck k  und Tab
17 40 15 0 7742. 3
20 40 22 0 7742. 324      Druck l_i und Tab
21 40 16 0 7743. 2
22 40 22 0 7743. 223      Druck m_i und Tab
23 40 75 4 7773.          S : = S + S
24 40 21 0 7773. 267      Druck S'
25 41 54 0 0003.
```

Wenn der interpretierend abzuarbeitende Maschinenbefehl einen unbedingten Sprung oder einen bedingten Sprung, dessen Bedingung erfüllt ist, darstellt, wird das Entschlüsselungsprogramm verlassen. Damit ist ein Übergang zur direkten verdrahteten Arbeitsweise möglich. Diese wird mit dem durch die Adresse des Sprungbefehls festgelegten Maschinenbefehl fortgesetzt. Der Übergang von einem reinen Maschinenprogramm zur interpretierenden Arbeitsweise ist u. a. folgendermaßen möglich:

a) Der Sprungbefehl (interpretierend abzuarbeitender Maschinenbefehl) sei in der Zelle b gespeichert. Wenn als letzter Befehl des Maschinenprogramms ein unbedingter Sprung zur Zelle 1401 bzw. 1402 programmiert ist (00 1401 000 / 00 1402 000), wird die interpretierende Arbeitsweise mit dem in der Zelle b + 1 gespeicherten Befehl fortgesetzt. [6])

   Bei einem Sprung nach 1401 wird der Inhalt von AC vorher ins Register R transportiert.

   Beispiel 51:

interpretierende Abarbeitung		verdrahtete Arbeitsweise
.		
.		
.		
b — 2 -------		
b — 1 -------		
b + 0 00 $a_1a_2a_3a_4$ 000	⟶	a + 0 ----
		a + 1 ----
		.
		.
		.
b + 1 -------	⟵	a + n 00 1401 000.
b + 2 -------		(a + n 00 1402 000.)
.		
.		
.		

b) Die interpretierende Arbeitsweise wird mit dem Befehl in der Zelle a + n fortgesetzt, wenn als letzter Befehl des Maschinenprogramms in der Zelle a + n — 1 ein spezieller Rücksprungbefehl „00 1443 025" programmiert ist. [7])

   Beispiel 52:

.		
.		
b — 2	------	interpretierende Arbeitsweise
b — 1	------	
b + 0	00 $a_1a_2a_3a_4$ 000	
a + 0	---	Maschinenprogramm (direkte verdrahtete Arbeitsweise)
a + 1	---	
.		
.		
.		
a + n — 1	00 1443 025.	
a + n	------	interpretierende Arbeitsweise
.		
.		
.		

[6]) Eine Voraussetzung für die Fortsetzung der interpretierenden Arbeitsweise ist, daß durch die Abarbeitung des Maschinenprogramms die Inhalte der Zellen 7754, 7755, 7770, 7673, 7767 nicht zerstört werden.

[7]) Nach einem Sprung ins Entschlüsselungsprogramm werden die Zellen 7754, 7755, 7767, 7770, 7673 neu aufgebaut.

6. Hilfsprogramme

Die Hilfsprogramme werden für die Testung von Programmen benötigt. Sie können auf beliebig freie Speicherplätze gebracht und mit Hilfe der 1. Kennzeichen $K_1 = D$ und $K_1 = B$ ausgewählt werden. Die Programme sind relativ programmiert. Bei der Abarbeitung von getesteten Programmen werden die Hilfsprogramme nicht benötigt, so daß der von ihnen belegte Speicherplatz anderweitig verwendet werden kann.

6.1. Druck von Pseudozähler und Inhalt des Registers R (B0/B1)

Bei Unterbrechung eines interpretierend abzuarbeitenden Programms (z. B. Überlauf) kann es von Vorteil sein, den Stand des Pseudozählers bzw. den Inhalt des Registers R zu kennen. Da keine direkte Anzeigemöglichkeit existiert, ist ein entsprechendes Druckprogramm bereitgestellt.

Der Stand des Pseudozählers wird als oktale Adresse $a_1a_2a_3a_4$ ausgegeben. Der Inhalt von R wird in Form von max. 11 Triaden bei der Eingabe von B0 bzw. als Dezimalzahl oder Externbefehl bei der Eingabe von B1 ausgedruckt.

Eingabebilder:	B0	Druck von ⟨R⟩ in Triaden
	B1	Druck von ⟨R⟩ als ganze Dezimalzahl oder Externbefehl

Nach der Ausgabe steht der Automat wieder im „Eingabeprogramm" und ist somit zur Entgegennahme eines neuen 1. Kennzeichens K_1 bereit. Die Kombinationen B0 und B1 können auch in beliebiger Reihenfolge hintereinander ausgeführt werden.

6.2. Fortsetzung eines unterbrochenen interpretierend arbeitenden Programmteiles (B3)

Beim Ansprung eines interpretierend abzuarbeitenden Programmteiles mit Hilfe des 1. Kennzeichens $K_1 = 5$

$$5\ a_1a_2a_3a_4\ \text{WM}\ b_1b_2b_3b_4\ \text{WM}$$

kann durch die Angabe der Adresse $b = b_1b_2b_3b_4 \neq 0$ ein laufendes Programm gestoppt werden (s. Abschnitt 2.2.5.). Das Programm wird **nach** der Abarbeitung des in der Zelle b gespeicherten Befehls unterbrochen. Der Automat steht dabei im Eingabeprogramm und ist zur Entgegennahme eines neuen K_1 bereit. Anschließend ist es möglich, z. B. den Stand des Pseudozählers und den Inhalt des Registers R (Hilfsprogramme (B0/B1) oder die Inhalte von bestimmten Zellen (Hilfsprogramme D3/D4) auszuschreiben.

Durch die Eingabe der Kombination

$$\text{B3}\ b_1b_2b_3b_4\ \text{WM}$$

ist eine Fortsetzung des Programms möglich. Die Adresse b hat dabei dieselbe Bedeutung wie bei $K_1 = 5$.

Eine Unterbrechung ist nicht möglich, wenn der in der Zelle b gespeicherte Befehl einen unbedingten Sprung (z. B. Externbefehle $S_1S_2 = 42/43/57$) oder einen bedingten Sprungbefehl mit erfüllter Sprungbedingung (z. B. Externbefehle $S_1S_2 = 31/32/41/60/61/62$) darstellt.

Außerdem ist bei dem Befehl „Eingabe von Dezimalzahlen im Satz" ($S_1S_2 = 21$) eine Unterbrechung bei der 8-Kanal-Version nur bei dem Schlußzeichen Satzmarke „SM" möglich.

Bei allen interpretierend abzuarbeitenden Maschinenbefehlen, die nicht aus dem Entschlüsselungsprogramm herausführen, ist eine Unterbrechung möglich.

Erfolgt ein Übergang zwischen interpretierender Abarbeitung und direkter verdrahteter Arbeitsweise entsprechend der im Abschnitt 5. unter a) beschriebenen Variante, ist nach der Rückkehr ins Entschlüsselungsprogramm – vor der Abarbeitung des Befehls in Zelle b + 1 – eine Unterbrechung möglich.

6.3. Druck des Inhaltes der Zelle 7777 (D0/D1)

Mit Hilfe der Eingriffstasten und -schalter kann der Inhalt des Maschinenakkus zum Speicherplatz 7777 (oktal) gebracht werden.

Folgende Kombintion ist notwendig: HGÜE

Der Inhalt von 7777 kann nun seinerseits entweder als Oktalzahl (Triadenschreibweise) bei der Eingabe von D0 bzw. als Dezimalzahl oder Externbefehl bei der Eingabe von D1 ausgedruckt werden.

Eingabebilder: D0 Druck von ⟨7777⟩ in Triaden
D1 Druck von ⟨7777⟩ als ganze Dezimalzahl oder Externbefehl

Nach der Ausgabe steht der Automat im „Eingabeprogramm". Die Kombinationen D0 und D1 können auch in beliebiger Reihenfolge hintereinander ausgeführt werden.

6.4. Bereichsausschreiben (D3/D4)

Durch diese Hilfsprogramme kann ein Speicherbereich, der durch eine Anfangsadresse $a = a_1a_2a_3a_4$ und eine Endadresse $e = e_1e_2e_3e_4$ festgelegt wird, in einer Liste ausgeschrieben werden. Die Adresse des jeweiligen Speicherplatzes wird mit ausgedruckt. Der Ansprung der Programme erfolgt mittels der Kombinationen D3 und D4.

Im Programm D3 wird der Zelleninhalt in Form von max. 11 Triaden ausgeschrieben.

Im Programm D4 wird bei der Ausgabe des Zelleninhaltes zwischen einem Externbefehl und einer Dezimalzahl unterschieden.

Eingabebilder: D3 $a_1a_2a_3a_4$ WM $e_1e_2e_3e_4$ WM

Eingabebilder: D4 $a_1a_2a_3a_4$ WM $e_1e_2e_3e_4$ WM

Führende Nullen können bei der Eingabe der Adressen unterdrückt werden. Eine Adresse ist durch „WM" abzuschließen. Soll nur der Inhalt einer Zelle ausgedruckt werden, genügt bei der Adresseneingabe die Darstellung „$a_1a_2a_3a_4$ WM WM" ($b = 0$). Hat man sich während der Eingabe einer Adresse vertippt, kann man nach Betätigen der Taste „ZR" diese neu eingeben. Nach der Ausgabe ist der Automat zur Entgegennahme einer neuen Anfangsadresse bereit.

Das Programm kann als Maschinenunterprogramm verwendet werden. Dabei sind folgende Festlegungen zu beachten (das Programm sei ab Zelle z gespeichert):

$z + 0$	Einsprung für Ausschreibungen analog D3
$z + 1$	00 $a_1a_2a_3a_4$ 000 Versorgungszelle für a
$z + 2$	00 $e_1e_2e_3e_4$ 000 Versorgungszelle für e
$z + 3$	Einsprung für Ausschreiben analog D4

6.5. Bereichsausstanzen (D8)

Das Programm kann durch D8 angesprungen werden. Eine Anfangsadresse a und eine Endadresse e legen den auszustanzenden Speicherbereich fest.

Die Anfangsadresse a wird als 1. LA, versehen mit den Eingabekennzeichen $K_1 = 2$ und $K_2 = 1$ sowie dem Schlußzeichen WM, ausgestanzt. Die Inhalte der Speicherzellen werden als Oktalzahlen (max. 11 aufeinanderfolgende Triaden), versehen mit $K_1 = K_2 = 0$ und dem Schlußzeichen „WM" (8-Kanal: Tab; 5-Kanal: „."), ausgestanzt. Zum Schluß wird das Lochbandschlußzeichen „/" ausgegeben.

Ein so erstelltes Lochband kann mit Hilfe des „Eingabeprogramms" wieder in den Rechner eingelesen werden.

Eingabebild: D8 $a_1a_2a_3a_4$ WM $e_1e_2e_3e_4$ WM

Führende Nullen können bei der Eingabe der Adressen unterdrückt werden. Eine Adresse ist durch „WM" abzuschließen.

Soll nur der Inhalt einer Zelle ausgestanzt werden, genügt bei der Adresseneingabe die Darstellung „$a_1a_2a_3a_4$ WM WM" ($b = 0$). Hat man

sich während der Eingabe einer Adresse vertippt, kann man diese nach Betätigen der Taste „ZR" neu eingeben. Nach dem Stanzen steht der Automat im „Eingabeprogramm". Das Programm kann als Maschinenunterprogramm verwendet werden. Dabei sind folgende Festlegungen zu beachten (das Programm sei ab Zelle z gespeichert):

z + 0	Einsprung	
z + 1	00 $a_1a_2a_3a_4$ 000	Versorgungszelle für a
z + 2	00 $e_1e_2e_3e_4$ 000	Versorgungszelle für e

6.6. Vergleichsprogramm (D5)

Mit Hilfe dieses Programms können bereits abgespeicherte Informationen zur Kontrolle mit der Sollinformation verglichen werden. Das Vergleichsprogramm speichert die eingegebenen Informationen nicht ab, sondern vergleicht sie mit den zugehörigen, bereits abgespeicherten Informationen. Stimmen die Worte nicht überein, werden die Adresse des Speicherplatzes, bei dem der Unterschied auftrat, der Inhalt des Speicherplatzes und die neu eingegebene Information in Triaden ausgedruckt. Nach dem Ansprung des Programms ist durch die Angabe der Ziffern 0 (AW0), 1 (AW1) bzw. 2 (AW2) zu entscheiden, über welchen Kanal die Informationen eingelesen werden sollen. Vor dem Vergleichen sind die Leitadressen einzugeben. Dies kann auch vor dem Ansprung des Vergleichsprogramms erfolgen. Die Eingabe einer Relativadresse vor Einsprung ins Vergleichsprogramm ist wirkungslos.

Die Eingabesteuerung erfolgt durch das „Eingabeprogramm" (Abschnitt 2.2.).

Nach dem Einlesen eines 1. Kennzeichens $K_1 = 8$ werden Dezimalzahlen entsprechend der im Abschnitt 2.2.7. dargestellten Organisation eingelesen und verglichen. (Die Fehleranzeige erfolgt in Triaden.)

Die Eingabe von Sofortbefehlen ($K_1 = 1, 5$) ist verboten.

Bemerkung: Beim Vergleichsprogramm wird jedes eingelesene Wort mit dem Inhalt des entsprechenden Speicherplatzes verglichen. Daher werden z. B. auch solche Informationen als fehlerhaft interpretiert, die erst nachträglich – z. B. durch die Eingabe von „Irrung-Satz" – korrigiert werden.

Eingabebild:	Eingabe der Hilfsadressen	D5	0	Tastatur
			1	Leser 1
			2	Leser 2

Bemerkung: Die Hilfsprogramme sind auf einem Lochband zusammengefaßt.

Bei Verwendung des Bereichsausschreibens und des Bereichsausstanzens als Unterprogramme ergeben sich folgende Einsprünge (Abspeicherung der Programme ab z):

Bereichsausschreiben:

z + 140 Einsprung für Ausschreiben analog D3
z + 141 Versorgungszelle für a
z + 142 Versorgungszelle für e
z + 143 Einsprung für Ausschreiben analog D4

Bereichsausstanzen:

z + 0 Einsprung
z + 1 Versorgungszelle für a
z + 2 Versorgungszelle für e

Bei Eingabe des Lochbandes muß die Anfangsadresse z als 1. und 2. Leitadresse gegeben werden:

a) 21 z WM
b) 22 z WM
c) Eingabe des Lochbandes

7. Anfertigen eines Lochbandes und Korrektur von Eingabefehlern

7.1. Anfertigen eines Lochbandes und Korrektur von Lochfehlern

Als Eingabemedium wird beim C 8205 ein 8-Kanal- bzw. 5-Kanal-Lochband verwendet. Die Codierung der Information erfolgt für das vorliegende System „FIPS I" im „Externen R 300-Code" bzw. im Telegraphenalphabet Nr. 2 (FS Code)".

Zum Anfertigen eines Lochbandes kann jedes lochbanderzeugende Gerät, das im R 300-Code bzw. im FS-Code ausgelegt ist, verwendet werden.

Ablochen eines Programms – Eingabe mit Kennzeichen gemäß Abschnitt 2.2.

Die Befehle, Konstanten usw. werden unmittelbar vom Programmformular abgelocht. Dabei ist bei dem ersten Kennzeichen K_1 zu beginnen. Danach schließen sich ohne Zwischenraum ein eventuelles zweites Kennzeichen K_2 und die eigentliche Information einschließlich WM an.

Die Adresse der Zelle (relativ oder absolut), in die eine Information gespeichert werden soll, wird nur bei Unterbrechung der Abspeicherung in natürlicher Reihenfolge abgelocht (vgl. Abschnitt 2.2.1.2.).

Bei der Eingabe führen alle Zeichen eines Codes, die nicht als erstes Kennzeichen definiert sind, aber als solche auf einem Lochband stehen, zur Eingabe eines neuen K_1 zurück (s. Abschnitt 2.2.). Sie werden überlesen. Dies betrifft im R 300-Code auch die Zeichen WR | WRZL/SM | ZL | GB | KB und im FS-Code die Zeichen WR | ZL | Transportlochung | Bu | Zi/Zei. Zur Gliederung des Lochbandes kann z. B. das Zeichen „ZR" verwendet werden (vgl. Abschnitt 2.2.10.). Es muß dann nach dem Ablochen eines vollständigen Wortes gegeben werden.

Als Lochbandvorspann kann ein Zeichen verwendet werden, das als 1. Kennzeichen überlesen wird, z. B. ZR | Irrung-Zeichen (8-Kanal)[8]) | Bu (5-Kanal) | Transportlochung[9]).

Ablochen von Dezimalzahlen – Eingabe ohne Kennzeichen

Die Dezimalzahlen können entsprechend ihrer üblichen Schreibweise abgelocht werden. Als Schlußzeichen ist WM zu geben (vgl. Abschnitte 2.2.7., 3.2.39., 3.2.40.).

Zur Gliederung des Lochbandes kann „ZR" verwendet werden. Im R 300-Code wird das Zeichen „WR", wenn es zwischen zwei Dezimalzahlen gelocht ist, überlesen. Dagegen wird das Zeichen „Zeilenvorschub" als Ziffer interpretiert. Bei seinem Ablochen muß eine entsprechende Korrektur vorgenommen werden (s. Fehlerkorrektur).

Als Lochbandvorspann können die Zeichen ZR | Irrung-Zeichen oder Transportlochung verwendet werden.

Im FS-Code werden die Zeichen Transportlochung | WR | ZL | Bu | Zi/Zei überlesen, wenn sie zwischen zwei Dezimalzahlen gelocht sind. Als Lochbandvorspann können die Zeichen Transportlochung | ZR und Bu verwendet werden.

Fehlerkorrektur

Eine Fehlerkorrektur ist prinzipiell immer dadurch möglich, daß der falsche Teil aus dem Lochband herausgeschnitten und der richtige Teil eingefügt wird. Darüber hinaus gibt es folgende Möglichkeiten:

a) **Innerhalb der Zeichen eines Wortes ist ein falsches Symbol gelocht – Das nächste Wort ist noch nicht gestanzt.**

   Der Fehler kann korrigiert werden, indem alle bereits gestanzten Zeichen des Wortes zugelocht werden. Im R 300-Code ist dies durch

---

[8]) Das Zeichen „Irrung-Zeichen" wird vom Leser als nicht vorhandenes Zeichen interpretiert.

[9]) Im R 300-Code wird „Transportlochung" als Vorspann überlesen. Sonst führt es zur Paritätskontrolle.

„Irrung-Zeichen", im FS-Code durch „Bu" möglich. Danach ist das Wort noch einmal, von vorn beginnend, einzugeben (bei Eingabe mit Kennzeichen mit K_1 beginnen).

b) **Innerhalb der Zeichen eines Wortes ist ein falsches Symbol gelocht – Das letzte Zeichen des Wortes ist noch nicht gelocht.**

Wenn das fehlerhafte Symbol den Maschinenbefehl „Eingabe-Halt" nicht in einen Sprungbefehl umstellt[10]), können durch Lochen von „ZR" die bereits gestanzten Zeichen eines Wortes gelöscht werden. Das Wort ist anschließend richtig zu wiederholen. (Bei Eingabe mit Kennzeichen beginnend bei K_1). Wird beim Ablochen im R 300-Code der Fehler unmittelbar nach dem Stanzen des fehlerhaften Symbols erkannt, kann dieses Zeichen durch „Irrung-Zeichen" überlocht werden. Danach ist mit dem Lochen des richtigen Zeichens fortzufahren[8]). Eventuell ergibt sich eine Korrekturmöglichkeit durch Ergänzungslochungen innerhalb des fehlerhaften Zeichens.

c) **Der Fehler wird bemerkt, nachdem bereits mehrere Worte abgelocht sind.**

Bei Eingabe mit Kennzeichen ist die Relativadresse des fehlerhaften Wortes und daran anschließend die Sollinformation abzulochen.[11]) Mit Hilfe der Relativadresse wird ein neuer Speicherbefehl aufgebaut (vgl. Abschnitt 2.2.1.2.). Bei der Eingabe wird das bereits abgespeicherte fehlerhafte Wort mit der richtigen Information überschrieben. Auf diese Art kann auch ein Korrekturlochband erstellt werden.

Bei der Eingabe von Dezimalzahlen im Satz kann ein fehlerhafter Satz gelöscht werden (s. Abschnitte 2.2.7. und 3.2.40.).

7.2. Eingabe über Tastatur

Die Eingaben über die Tastatur und die Lochbandleser sind gleichberechtigt. Die im Abschnitt 7.1. gemachten Ausführungen gelten daher sinngemäß auch für die Eingabe über die Tastatur des Bedienungspultes bzw. die des Schreibwerkes. Darüber hinaus kann bei der Eingabe mit Kennzeichen eine Fehlerkorrektur immer durch eine Rückkehr ins Eingabeprogramm (Kombination HGR) eingeleitet werden.

---

[10]) Ziffern stellen das „Eingabe-Halt" nicht in einen Sprungbefehl um.

[11]) Bei absoluter Programmierung kann natürlich die Absolutadresse angegeben werden.

8. Bedienungshinweise

8.1. Ansprung des „Eingabeprogramms"

Das Eingabeprogramm kann auf der Zelle Null mit Hilfe der Eingriffstasten und -schalter

a) Schalter „H–R" H drücken
b) Taste G
c) Schalter „H–R" R drücken

angesprungen werden. Nach der Kombination

HGR

wartet der Automat auf die Eingabe eines ersten Kennzeichens K_1.

8.2. Programmeingabe

Die Eingabe eines Programms sei an folgendem Beispiel dargestellt (s. auch Abschnitt 2.2.):

In einem Hauptprogramm werden zwei Unterprogramme, die mit 1. UP und 2. UP bezeichnet sind, verwendet. Das 1. UP benutzt außerdem das 2. UP als UUP. Das HP ist relativ bezüglich der 1., 2. und 3. LA programmiert. Das 1. UP ist relativ bezüglich der 1. und 2. LA programmiert und das 2. UP relativ bezüglich der 1. LA. Das Hauptprogramm HP soll ab Zelle 2000, das 1. UP ab Zelle 2400 und das 2. UP ab Zelle 2500 gespeichert werden. Das HP ist relativ auf dem Speicherplatz 4 anzuspringen.

Nach dem Ansprung des Eingabeprogramms durch HGR können z. B. die Hilfsadressen, falls sie nicht auf dem Lochband stehen, und die Programme folgendermaßen eingegeben werden:

a) Eingabe des 2. UP:

Eingabe der 1. LA	21 2500.
Anwahl eines Lesers und Programmeingabe	HK_1/K_2R

b) Eingabe des 1. UP:

Anwahl der Tastatureingabe	HGR
Eingabe der 1. LA	21 2400.
Eingabe der 2. LA (1. LA des 2. UP)	22 2500.
Anwahl eines Lesers und Programmeingabe	HK_1/K_2R

c) Eingabe des HP:

Anwahl der Tastatureingabe	HGR
Eingabe der 1. LA	21 2000.
Eingabe der 2. LA (1. LA des 1. UP)	22 2400.
Eingabe der 3. LA (1. LA des 2. UP)	23 2500.
Anwahl eines Lesers und Programmeingabe	HK_1/K_2R

d) Ansprung des Hauptprogramms:

Anwahl der Tastatureingabe	HGR
Handelt es sich um ein Programm, das sich auf die interpretierende Abarbeitung bezieht	5 2004. .
Handelt es sich um ein Maschinenprogramm	11 4 000. oder 10 2004 000.

8.3. Korrektur von fehlerhaften Plätzen

Als Beispiel mögen auf den Speicherzellen 2010, 2011 und 2127 falsche Informationen stehen. Das zugehörige relativ geschriebene Programm wurde ab 2000 gespeichert. Die Korrektur kann mit Hilfe der Relativadresse durchgeführt werden.

a) Eingabe der 1. LA	21 2000.
b) Eingabe der Relativadresse	20 10.
c) Eingabe der Sollinformation der Zelle 10	

d) Eingabe der Sollinformation der Zelle 11

e) Eingabe der Relativadresse 20 127.

f) Eingabe der Sollinformation der Zelle 127

8.4. Druck des Maschinenbefehlszählers und des Maschinenakkumulators

Nach Programmunterbrechungen (z. B. Überlauf) kann es von Vorteil sein, den Stand des Maschinenbefehlszählers und den Inhalt des Maschinenakkus zu kennen. Da der C 8205 hierfür keine direkte Anzeigemöglichkeit besitzt, ist im FIPS I ein entsprechendes Druckprogramm bereitgestellt. In einem ersten Schritt kann der Inhalt des Maschinenakkus durch die Kombination HGÜE (⟨7777⟩ : = ⟨AC⟩) gerettet werden. Durch die Tasten- und Schalterkombination

GM↓EM↑R

ist anschließend das Druckprogramm anzuspringen. Ausgegeben wird die oktale Adresse des nächsten abzuarbeitenden Befehls. (Das Druckbild umfaßt 11 Triaden. Die oktale Adresse ist in den Oktalstellen t_3 bis t_6 angegeben.) Nach dem Druck steht der Automat im Eingabeprogramm. Durch die Eingabe von D0/D1 (Abschnitt 6.3.) kann der Inhalt von 7777 ausgegeben werden.

8.5. Druck des Inhaltes einer Zelle

a) Der Inhalt eines Speicherplatzes $a = a_1a_2a_3a_4$ kann durch einen Sofortlesebefehl in Maschinensprache

10 $a_1a_2a_3a_4$ 211.

in den Maschinenakku und durch anschließendes Drücken von HGÜE nach der Zelle 7777 transportiert werden.

Nach dem erneuten Ansprung des Eingabeprogramms (Drücken von GR), kann der Inhalt von 7777 durch D0/D1 ausgeschrieben werden.

b) Der Inhalt der Zelle $a = a_1a_2a_3a_4$ kann auch mit Hilfe des „Bereichsausschreibens" (s. Abschnitt 6.4.) ausgedruckt werden.

Eingabebild: (HGR) D $\begin{smallmatrix}3\\4\end{smallmatrix}$ $a_1a_2a_3a_4$. .

Tabelle 1

0	#	00
1	<	01
2	=	02
3	≈	03
4	[	04
5	>	05
6	¬	06
7	≋	07
8	~	10
9	]	11
:	(	15
+	═	20
.	¦	33
—	△	40
,	;	60
/	)	61
%	•	74
'	?	73
a	A	21
b	B	22
c	C	23
d	D	24
e	E	25
f	F	26
g	G	27
h	H	30
i	I	31
j	J	41
k	K	42
l	L	43
m	M	44
n	N	45
o	O	46
p	P	47
q	Q	50
r	R	51
s	S	62
t	T	63
u	U	64
v	V	65
w	W	66
x	X	67
y	Y	70
z	Z	71
	ZWR	12
	Tab	75
	WRZL	32
	WR	52
	ZL	72
	GB	37
	KB	57

V-14-6 5 485 S 617/72

VEB RECHENELEKTRONIK
MEININGEN/ZELLA-MEHLIS
IM VEB KOMBINAT
ZENTRONIK

PROGRAMMIERUNG UND BEDIENUNG DES DIGITALEN KLEINRECHNERS C 8205

HEFT 5

FESTKOMMA-INTERPRETIERSYSTEM FIPS I

VEB Rechenelektronik Meiningen / Zella-Mehlis
im VEB KOMBINAT ZENTRONIK
DDR - 606 Zella-Mehlis
Deutsche Demokratische Republik

Exporteur:
BÜROMASCHINEN-EXPORT GmbH.
DDR - 108 Berlin 8
Friedrichstraße 61